

# Appendix 2: Employment Variable Creation

## Contents

- Introduction
  - emp\_begin.sas program
  - Number of Weeks Worked during 19xx/20xx
  - Number of Weeks Worked since Last Interview
  - Number of Weeks Worked Age 14-19, Weeks Worked at All Jobs Since Age 20, and Weeks Worked at Employee-Type Jobs Since Age 20
  - Weeks Worked at Employee Job #x during 19xx/20xx
  - Total Tenure at Job #x as of the Survey Date
  - Total Hours Worked in 19xx/20xx
  - Total Hours Worked Age 14-19, Hours Worked at All Jobs Since Age 20, and Hours Worked at Employee-Type Jobs Since Age 20
  - Number of Jobs Held during 19xx/20xx
  - Total Number of Jobs Held Age 14-19, Number of Employee Jobs Held since Age 20, and Number of All Jobs Held since Age 20
  - Job Length Less Than 13 Weeks
- 

## Introduction

**Collection of Employment Data.** The employment sections of the NLSY97 questionnaire are somewhat complex. Before beginning analysis, researchers must understand the structure of each round's questionnaire, particularly the way in which jobs are classified as employee, freelance, or self-employment. It is important to note that this classification depends in part on the survey round and the respondent's age. In rounds 1 and 2, employee jobs were recorded in the first part of the YEMP section, administered only to respondents age 14 or older as of the interview date. The second part of the YEMP section collected information about freelance jobs of respondents age 14 and older and all jobs of respondents age 12 or 13 (the implicit assumption being that respondents younger than 14 are not likely to hold employee jobs). If the respondent was at least 16 years old and made at least \$200/week in a freelance job, the job was classified as self-employment and an extra series of questions was asked during the freelance section.

In round 3, all respondents were at least age 14 by the interview date, so the age restriction for employee jobs was no longer necessary. The structure of the section remained largely the same, with a division between employee and freelance jobs. Self-employment was classified in the same way as in the earlier rounds.

In round 4 the section was redesigned. Respondents born in 1980-82 (who were mostly age 18 and older when the round 4 field period began) were asked about employee jobs and self-employment at the same time. In addition, the minimum income requirement from the freelance section no longer applied; jobs could be classified as self-employment regardless of earnings. However, respondents born in 1983-84 (who were mostly age 16 or 17 when the round 4 field period began) continued to describe employee and freelance jobs separately. Data on self-employment jobs were still collected in the freelance section, and freelance jobs still had to meet the income criteria to qualify as self-employment. The same pattern was used in round 5.

The redesign of the employment section has important implications for created employment variables. In rounds 1-3, all of the created employment variables were based only on employee-type jobs. So, for example, the variable "Weeks Worked during Calendar Year 1999" counted only the weeks worked by a respondent at a regular employee-type job. If the respondent also reported self-employment in a lawn care business, the weeks spent working at that job were not counted in the created variable.

In round 4, when older respondents reported both employee-type and self-employed jobs in the same series of variables, this approach was reconsidered. For rounds 4 and 5, older respondents now have three versions of most created variables. The first version, identified by the suffix "ET" in the question name, includes only employee-type jobs. The second version, the "SE" variables, includes only self-employed jobs reported by respondents born in 1980-83 in the regular employment section **during round 5** (and similarly for respondents born in 1980-82 in round 4). These variables do **not** include freelance jobs or self-employment reported by younger respondents in the freelance jobs section in rounds 4 and 5, and they do **not** include freelance jobs or self-employment reported in rounds 1-3 by any respondent regardless of age. Finally, the variables for all jobs include both employee-type jobs and self-employment reported during round 5 for respondents born in 1980-83 but only include employee-type jobs for respondents born in 1984 (similarly, in round 4, these variables reported all jobs for respondents born in 1980-82 and only employee-type jobs for respondents born in 1983-84). These last variables are identified with the suffix "ALL" in the question name.

Respondents' ages varied widely in round 4, when self-employed jobs were first recorded as part of the regular employment section, and some respondents reported employment over several years if they missed a round of interviewing. To simplify the creation of the employment variables, survey staff included only the self-employment job information starting on January 1 of the year the respondent turned 18. For example, consider a respondent who was 20 years old on his round 4 interview date in April 2001 and had not been interviewed since round 1. He reports self-employment in a computer repair business beginning on his 17<sup>th</sup> birthday in March 1998 and continuous employment at a fast-food restaurant since his round 1 interview in 1997. The round 4 created employment variables would include information about the employee-type fast food job dating all the way back to 1997. However, the computer repair business would not be considered until January 1, 1999 (the first day of the year he turned 18). In other words, the variable "Weeks Worked in Calendar Year 1998" would count only the fast food job and the variable "Weeks Worked Any Job in Calendar Year 1999" would count both the fast food job and the repair business. Similarly, the new variable "Weeks R Was Self-Employed Year 1998" would have a value of -4, or valid skip (because the respondent was not yet 18), but the variable "Weeks R Was Self-Employed Year 1999" would report the weeks the respondent worked at the computer repair business in 1999. This approach, continued in round 5, will permit users to compare the employment variables across respondents in different rounds with confidence that the job types included are the same for all respondents of a given age.

**"Backreporters."** Occasionally respondents report a job in the current interview that started before the date of their last interview and should have been reported at that time. [Appendix 6](#) in this document contains a more complete description of the implications of these reports for the created employment event history variables. There are also a number of created employment variables detailed in this appendix. Nearly all of these variables use the information provided about employment previous to the date of last interview. The only exception is the set of CV\_WKSWK\_DLI variables, which reports the weeks worked since the previous interview date. The current round's variables would not include the backreported information in any case; the previous round's variables are **not** re-created to incorporate this new information.

For example, assume that Jane was interviewed in round 3 on April 15, 2000, and in round 4 on April

15, 2001. In the round 4 interview, she reports for the first time a job that started on April 1, 2000. The 2 weeks worked at that job before April 15, 2000, would **not** be reflected in any round's CV\_WKSWK\_DLI variables. However, those weeks **would** be counted in other variables. For example, CV\_WKSWK\_YR.00 would count all the weeks worked at any job in 2000, regardless of whether those weeks were reported in the round 3 or round 4 interview.

**Creation of Employment Variables.** A number of the created employment variables use the same program as input. The program in this section is referred to throughout the employment variables. For example, to create the "Weeks Worked at Employee Job #x during 19xx" variables, survey staff first run the program below titled "emp\_begin.sas" and then run the program included in the weeks worked section of this appendix.

### Emp\_begin.sas

This program calculates total weeks worked at each job for each respondent. It converts start and stop dates for jobs and within-job gaps to continuous week numbers, subtracts within-job gaps, and finally counts the total weeks worked. The variables listed here are those employed in any program that uses emp\_begin.sas as input; they may not be used in this initial program.

Name in Program	Question Name on CD	Name in Program	Question Name on CD
r1int_d	YINF-900_D	e381031	YEMP-38103.01
r1int_m	YINF-900_M	e381032	YEMP-38103.02
r1int_y	YINF-900_Y	e381033	YEMP-38103.03
birthdy	KEY!BDATE_D	e381051	YEMP-38105.01
birthmo	KEY!BDATE_M	e381052	YEMP-38105.02
birthyr	KEY!BDATE_Y	e381053	YEMP-38105.03
norcid	YNORCID	e599011	YEMP-59901.01
r2int_d	YINTDATE~D	e599012	YEMP-59901.02
r2int_m	YINTDATE~M	e599013	YEMP-59901.03
r2int_y	YINTDATE~Y	e599014	YEMP-59901.04
r3int_d	YINTDATE~D	e599015	YEMP-59901.05
r3int_m	YINTDATE~M	e599016	YEMP-59901.06
r3int_y	YINTDATE~Y	e599017	YEMP-59901.07
r4int_d	YINTDATE~D	e599018	YEMP-59901.08
r4int_m	YINTDATE~M	e880001	YEMP-88000.01
r4int_y	YINTDATE~Y	e880002	YEMP-88000.02
milflag1	YEMP_MILFLAG.01	e880003	YEMP-88000.03
milflag2	YEMP_MILFLAG.02	e880004	YEMP-88000.04
milflag3	YEMP_MILFLAG.03	e880005	YEMP-88000.05
milflag4	YEMP_MILFLAG.04	e880006	YEMP-88000.06
milflag5	YEMP_MILFLAG.05	e880007	YEMP-88000.07
milflag6	YEMP_MILFLAG.06	e885011	YEMP-88501.01
milflag7	YEMP_MILFLAG.07	e885012	YEMP-88501.02
milflag8	YEMP_MILFLAG.08	e885013	YEMP-88501.03
self1	YEMP_SELFEMP.01	e885014	YEMP-88501.04
self2	YEMP_SELFEMP.02	e885015	YEMP-88501.05
self3	YEMP_SELFEMP.03	e885016	YEMP-88501.06
self4	YEMP_SELFEMP.04	e984021	YEMP-98402.01
self5	YEMP_SELFEMP.05	e984022	YEMP-98402.02
self6	YEMP_SELFEMP.06	e984023	YEMP-98402.03
self7	YEMP_SELFEMP.07	e984024	YEMP-98402.04

self8	YEMP_SELFEMP.08	e984025	YEMP-98402.05
stardy1	YEMP_STARTDATE.01~D	e984026	YEMP-98402.06
starmo1	YEMP_STARTDATE.01~M	e984031	YEMP-98403.01
staryr1	YEMP_STARTDATE.01~Y	e984032	YEMP-98403.02
stardy2	YEMP_STARTDATE.02~D	e984033	YEMP-98403.03
starmo2	YEMP_STARTDATE.02~M	e984291	YEMP-98429.01
staryr2	YEMP_STARTDATE.02~Y	e984292	YEMP-98429.02
stardy3	YEMP_STARTDATE.03~D	e984293	YEMP-98429.03
starmo3	YEMP_STARTDATE.03~M	e1002311	YEMP-100231.01.02
staryr3	YEMP_STARTDATE.03~Y	e1002312	YEMP-100231.02.02
stardy4	YEMP_STARTDATE.04~D	e1002321	YEMP-100232.01.03
starmo4	YEMP_STARTDATE.04~M	e1002322	YEMP-100232.02.03
staryr4	YEMP_STARTDATE.04~Y	GAP1	YEMP-101500.01
stardy5	YEMP_STARTDATE.05~D	GAP2	YEMP-101500.02
starmo5	YEMP_STARTDATE.05~M	GAP3	YEMP-101500.03
staryr5	YEMP_STARTDATE.05~Y	GAP4	YEMP-101500.04
stardy6	YEMP_STARTDATE.06~D	GAP5	YEMP-101500.05
starmo6	YEMP_STARTDATE.06~M	GAP6	YEMP-101500.06
staryr6	YEMP_STARTDATE.06~Y	GAP7	YEMP-101500.07
stardy7	YEMP_STARTDATE.07~D	GAP8	YEMP-101500.08
starmo7	YEMP_STARTDATE.07~M	BGDY1_1	YEMP-102200.01.01~D
staryr7	YEMP_STARTDATE.07~Y	BGMO1_1	YEMP-102200.01.01~M
stardy8	YEMP_STARTDATE.08~D	BGYR1_1	YEMP-102200.01.01~Y
starmo8	YEMP_STARTDATE.08~M	BGDY2_1	YEMP-102200.02.01~D
staryr8	YEMP_STARTDATE.08~Y	BGMO2_1	YEMP-102200.02.01~M
stopdy1	YEMP_STOPDATE.01~D	BGYR2_1	YEMP-102200.02.01~Y
stopmo1	YEMP_STOPDATE.01~M	BGDY3_1	YEMP-102200.03.01~D
stopyr1	YEMP_STOPDATE.01~Y	BGMO3_1	YEMP-102200.03.01~M
stopdy2	YEMP_STOPDATE.02~D	BGYR3_1	YEMP-102200.03.01~Y
stopmo2	YEMP_STOPDATE.02~M	BGDY4_1	YEMP-102200.04.01~D
stopyr2	YEMP_STOPDATE.02~Y	BGMO4_1	YEMP-102200.04.01~M
stopdy3	YEMP_STOPDATE.03~D	BGYR4_1	YEMP-102200.04.01~Y
stopmo3	YEMP_STOPDATE.03~M	BGDY5_1	YEMP-102200.05.01~D
stopyr3	YEMP_STOPDATE.03~Y	BGMO5_1	YEMP-102200.05.01~M
stopdy4	YEMP_STOPDATE.04~D	BGYR5_1	YEMP-102200.05.01~Y
stopmo4	YEMP_STOPDATE.04~M	BGDY1_2	YEMP-102400.01.02~D
stopyr4	YEMP_STOPDATE.04~Y	BGMO1_2	YEMP-102400.01.02~M
stopdy5	YEMP_STOPDATE.05~D	BGYR1_2	YEMP-102400.01.02~Y
stopmo5	YEMP_STOPDATE.05~M	BGDY1_3	YEMP-102400.01.03~D
stopyr5	YEMP_STOPDATE.05~Y	BGMO1_3	YEMP-102400.01.03~M
stopdy6	YEMP_STOPDATE.06~D	BGYR1_3	YEMP-102400.01.03~Y
stopmo6	YEMP_STOPDATE.06~M	BGDY1_4	YEMP-102400.01.04~D
stopyr6	YEMP_STOPDATE.06~Y	BGMO1_4	YEMP-102400.01.04~M
stopdy7	YEMP_STOPDATE.07~D	BGYR1_4	YEMP-102400.01.04~Y
stopmo7	YEMP_STOPDATE.07~M	BGDY1_5	YEMP-102400.01.05~D
stopyr7	YEMP_STOPDATE.07~Y	BGMO1_5	YEMP-102400.01.05~M
stopdy8	YEMP_STOPDATE.08~D	BGYR1_5	YEMP-102400.01.05~Y
stopmo8	YEMP_STOPDATE.08~M	BGDY1_6	YEMP-102400.01.06~D
stopyr8	YEMP_STOPDATE.08~Y	BGMO1_6	YEMP-102400.01.06~M
uid1	YEMP_UID.01	BGYR1_6	YEMP-102400.01.06~Y
uid2	YEMP_UID.02	BGDY2_2	YEMP-102400.02.02~D

uid3	YEMP_UID.03	BGMO2_2	YEMP-102400.02.02~M
uid4	YEMP_UID.04	BGYR2_2	YEMP-102400.02.02~Y
uid5	YEMP_UID.05	BGDY2_3	YEMP-102400.02.03~D
uid6	YEMP_UID.06	BGMO2_3	YEMP-102400.02.03~M
uid7	YEMP_UID.07	BGYR2_3	YEMP-102400.02.03~Y
uid8	YEMP_UID.08	BGDY3_2	YEMP-102400.03.02~D
r5int_d	YINTDATE~D	BGMO3_2	YEMP-102400.03.02~M
r5int_m	YINTDATE~M	BGYR3_2	YEMP-102400.03.02~Y
r5int_y	YINTDATE~Y	BGDY3_3	YEMP-102400.03.03~D
e200	YEMP-200A	BGMO3_3	YEMP-102400.03.03~M
e303_1	YEMP-303.01	BGYR3_3	YEMP-102400.03.03~Y
e303_2	YEMP-303.02	BGDY4_2	YEMP-102400.04.02~D
e303_3	YEMP-303.03	BGMO4_2	YEMP-102400.04.02~M
e303_4	YEMP-303.04	BGYR4_2	YEMP-102400.04.02~Y
e239011	YEMP-23901.01	EGDY1_1	YEMP-102700.01.01~D
e239012	YEMP-23901.02	EGMO1_1	YEMP-102700.01.01~M
e239013	YEMP-23901.03	EGYR1_1	YEMP-102700.01.01~Y
e239014	YEMP-23901.04	EGDY1_2	YEMP-102700.01.02~D
e239015	YEMP-23901.05	EGMO1_2	YEMP-102700.01.02~M
e239016	YEMP-23901.06	EGYR1_2	YEMP-102700.01.02~Y
e239017	YEMP-23901.07	EGDY1_3	YEMP-102700.01.03~D
e245011	YEMP-24501.01	EGMO1_3	YEMP-102700.01.03~M
e245012	YEMP-24501.02	EGYR1_3	YEMP-102700.01.03~Y
e245013	YEMP-24501.03	EGDY1_4	YEMP-102700.01.04~D
e245014	YEMP-24501.04	EGMO1_4	YEMP-102700.01.04~M
e245015	YEMP-24501.05	EGYR1_4	YEMP-102700.01.04~Y
e245016	YEMP-24501.06	EGDY1_5	YEMP-102700.01.05~D
e344021	YEMP-34402.01	EGMO1_5	YEMP-102700.01.05~M
e344022	YEMP-34402.02	EGYR1_5	YEMP-102700.01.05~Y
e344023	YEMP-34402.03	EGDY1_6	YEMP-102700.01.06~D
e344024	YEMP-34402.04	EGMO1_6	YEMP-102700.01.06~M
e344025	YEMP-34402.05	EGYR1_6	YEMP-102700.01.06~Y
e344026	YEMP-34402.06	EGDY2_1	YEMP-102700.02.01~D
e344031	YEMP-34403.01	EGMO2_1	YEMP-102700.02.01~M
e344032	YEMP-34403.02	EGYR2_1	YEMP-102700.02.01~Y
e344033	YEMP-34403.03	EGDY2_2	YEMP-102700.02.02~D
e344281	YEMP-34428.01	EGMO2_2	YEMP-102700.02.02~M
e344282	YEMP-34428.02	EGYR2_2	YEMP-102700.02.02~Y
e344283	YEMP-34428.03	EGDY2_3	YEMP-102700.02.03~D
e226101	YEMP-22610.01.02	EGMO2_3	YEMP-102700.02.03~M
e226102	YEMP-22610.02.02	EGYR2_3	YEMP-102700.02.03~Y
e226103	YEMP-22610.03.02	EGDY3_1	YEMP-102700.03.01~D
e226111	YEMP-22611.01.03	EGMO3_1	YEMP-102700.03.01~M
e226112	YEMP-22611.02.03	EGYR3_1	YEMP-102700.03.01~Y
e226113	YEMP-22611.03.03	EGDY3_2	YEMP-102700.03.02~D
e379021	YEMP-37902.01	EGMO3_2	YEMP-102700.03.02~M
e379022	YEMP-37902.02	EGYR3_2	YEMP-102700.03.02~Y
e379023	YEMP-37902.03	EGDY3_3	YEMP-102700.03.03~D
e379024	YEMP-37902.04	EGMO3_3	YEMP-102700.03.03~M
e379025	YEMP-37902.05	EGYR3_3	YEMP-102700.03.03~Y
e379026	YEMP-37902.06	EGDY4_1	YEMP-102700.04.01~D

e379027	YEMP-37902.07	EGMO4_1	YEMP-102700.04.01~M
e379041	YEMP-37904.01	EGYR4_1	YEMP-102700.04.01~Y
e379042	YEMP-37904.02	EGDY4_2	YEMP-102700.04.02~D
e379043	YEMP-37904.03	EGMO4_2	YEMP-102700.04.02~M
e379044	YEMP-37904.04	EGYR4_2	YEMP-102700.04.02~Y
e379046	YEMP-37904.06	EGDY5_1	YEMP-102700.05.01~D
e380001	YEMP-38000.01	EGMO5_1	YEMP-102700.05.01~M
e380002	YEMP-38000.02	EGYR5_1	YEMP-102700.05.01~Y
e380003	YEMP-38000.03	NUMGAP1	YEMP-103200.01
e380004	YEMP-38000.04	NUMGAP2	YEMP-103200.02
e380005	YEMP-38000.05	NUMGAP3	YEMP-103200.03
e380006	YEMP-38000.06	NUMGAP4	YEMP-103200.04
e38000F1	YEMP-38000F.01	NUMGAP5	YEMP-103200.05
e38000F2	YEMP-38000F.02		
e38000F3	YEMP-38000F.03		
e38000F4	YEMP-38000F.04		
e38000F5	YEMP-38000F.05		

/\*\*\*\*\* SECTION 1: Convert birth week, week turned age 14, week turned age 20,  
\*\*\*\*\*

\*\*\*\*\* and week of interview date into continuous weeks  
\*\*\*\*\*\*/

\*\* Calulate Age 14 year \*\*  
AGE14YR=birthyr+14;

\*\* Convert Age 14 Birthdate to week number \*\*  
\*\*\*Convert age14 month and day to total days (bbdays);

```
if birthmo>0 and birthdy>0 then do;
  if birthmo=1 then bbdays=birthdy;
  if birthmo=2 then bbdays=birthdy+31;
  if birthmo=3 then bbdays=birthdy+59;
  if birthmo=4 then bbdays=birthdy+90;
  if birthmo=5 then bbdays=birthdy+120;
  if birthmo=6 then bbdays=birthdy+151;
  if birthmo=7 then bbdays=birthdy+181;
  if birthmo=8 then bbdays=birthdy+212;
  if birthmo=9 then bbdays=birthdy+243;
  if birthmo=10 then bbdays=birthdy+273;
  if birthmo=11 then bbdays=birthdy+304;
  if birthmo=12 then bbdays=birthdy+334;
end;
```

\*\*\*Account for leap years;  
if age14yr=1980 or age14yr=1984 or age14yr=1988 or age14yr=1992 or age14yr=1996 or  
age18yr=2000 then do;  
 if birthmo>0 and birthdy>0 then do;

```
if birthmo=1 then bbdays=birthdy;  
if birthmo=2 then bbdays=birthdy+31;  
if birthmo=3 then bbdays=birthdy+60;  
if birthmo=4 then bbdays=birthdy+91;  
if birthmo=5 then bbdays=birthdy+121;  
if birthmo=6 then bbdays=birthdy+152;  
if birthmo=7 then bbdays=birthdy+182;  
if birthmo=8 then bbdays=birthdy+213;  
if birthmo=9 then bbdays=birthdy+244;  
if birthmo=10 then bbdays=birthdy+274;  
if birthmo=11 then bbdays=birthdy+305;  
if birthmo=12 then bbdays=birthdy+335;  
end;  
end;
```

\*\*\*Convert days into week numbers;

**\*\* Basic Formula: used throughout this program**

\*\* weekno=endweek{specific year}+ceil[(totdays+{# of days remaining in DEC})/7] \*\*

/\* Default age 14 week = 9999 \*/

age14wk=9999;

if age14yr>0 and bbdays>0 then do;

```

if age14yr=1993 then do; age14wk=678+ceil((bbdays+5)/7); end;
if age14yr=1994 then do; age14wk=730+ceil((bbdays+6)/7); end;
if age14yr=1995 then do; age14wk=783+ceil((bbdays)/7); end;
if age14yr=1996 then do; age14wk=835+ceil((bbdays+1)/7); end;
if age14yr=1997 then do; age14wk=887+ceil((bbdays+3)/7); end;
if age14yr=1998 then do; age14wk=939+ceil((bbdays+4)/7); end;
if age14yr=1999 then do; age14wk=991+ceil((bbdays+5)/7); end;
if age14yr=2000 then do; age14wk=1043+ceil((bbdays+6)/7); end
nd;
```

\*\* Calculate Age 18 year \*\*

AGE18YR=birthyr+18;

\*\* Convert Age 18 Birthdate to week number \*\*

\*\*\*Convert age18 month and day to total days (bbdays);

```

if birthmo>0 and birthdy>0 then do;
  if birthmo=1 then bbdays=birthdy;
  if birthmo=2 then bbdays=birthdy+31;
  if birthmo=3 then bbdays=birthdy+59;
  if birthmo=4 then bbdays=birthdy+90;
  if birthmo=5 then bbdays=birthdy+120;
  if birthmo=6 then bbdays=birthdy+151;
  if birthmo=7 then bbdays=birthdy+181;
  if birthmo=8 then bbdays=birthdy+212;
  if birthmo=9 then bbdays=birthdy+243;
  if birthmo=10 then bbdays=birthdy+273;
  if birthmo=11 then bbdays=birthdy+304;

```

```
if birthmo=12 then bbdays=birthdy+334;  
end;
```

\*\*\*Account for leap years;

```
if age18yr=1980 or age18yr=1984 or age18yr=1988 or age18yr=1992 or age18yr=1996 or  
age18yr=2000 then do;
```

```
if birthmo>0 and birthdy>0 then do;  
if birthmo=1 then bbdays=birthdy;  
if birthmo=2 then bbdays=birthdy+31;  
if birthmo=3 then bbdays=birthdy+60;  
if birthmo=4 then bbdays=birthdy+91;  
if birthmo=5 then bbdays=birthdy+121;  
if birthmo=6 then bbdays=birthdy+152;  
if birthmo=7 then bbdays=birthdy+182;  
if birthmo=8 then bbdays=birthdy+213;  
if birthmo=9 then bbdays=birthdy+244;  
if birthmo=10 then bbdays=birthdy+274;  
if birthmo=11 then bbdays=birthdy+305;  
if birthmo=12 then bbdays=birthdy+335;  
end;
```

```
end;
```

```
end;
```

\*\*\*Convert days into week numbers;

```
/* Default age 14 week = 9999 */
```

```
age18wk=9999;
```

```
if age18yr>0 and bbdays>0 then do;
```

```
if age18yr=1997 then do; age18wk=887+ceil((bbdays+3)/7); ag18jan=888; end;  
if age18yr=1998 then do; age18wk=939+ceil((bbdays+4)/7); ag18jan=940; end;  
if age18yr=1999 then do; age18wk=991+ceil((bbdays+5)/7); ag18jan=992; end;  
if age18yr=2000 then do; age18wk=1043+ceil((bbdays+6)/7); ag18jan=1044; end;  
if age18yr=2001 then do; age18wk=1096+ceil((bbdays+1)/7); ag18jan=1097; end;  
if age18yr=2002 then do; age18wk=1148+ceil((bbdays+2)/7); ag18jan=1149; end;  
end;
```

\*\* Calulate Age 20 week \*\*

```
AGE20YR=birthyr+20;
```

\*\* Convert Age 20 Birthdate to week number \*\*

```
***Convert age20 month and day to total days (bbdays);
```

```
if birthmo>0 and birthdy>0 then do;
```

```
if birthmo=1 then bbdays=birthdy;  
if birthmo=2 then bbdays=birthdy+31;  
if birthmo=3 then bbdays=birthdy+59;  
if birthmo=4 then bbdays=birthdy+90;  
if birthmo=5 then bbdays=birthdy+120;  
if birthmo=6 then bbdays=birthdy+151;  
if birthmo=7 then bbdays=birthdy+181;  
if birthmo=8 then bbdays=birthdy+212;  
if birthmo=9 then bbdays=birthdy+243;
```

```

if birthmo=10 then bbdays=birthdy+273;
if birthmo=11 then bbdays=birthdy+304;
if birthmo=12 then bbdays=birthdy+334;
end;

```

\*\*\*Account for leap years;

```

if age18yr=1980 or age18yr=1984 or age18yr=1988 or age18yr=1992 or age18yr=1996 or
age18yr=2000 then do;

```

```

if birthmo>0 and birthdy>0 then do;
  if birthmo=1 then bbdays=birthdy;
  if birthmo=2 then bbdays=birthdy+31;
  if birthmo=3 then bbdays=birthdy+60;
  if birthmo=4 then bbdays=birthdy+91;
  if birthmo=5 then bbdays=birthdy+121;
  if birthmo=6 then bbdays=birthdy+152;
  if birthmo=7 then bbdays=birthdy+182;
  if birthmo=8 then bbdays=birthdy+213;
  if birthmo=9 then bbdays=birthdy+244;
  if birthmo=10 then bbdays=birthdy+274;
  if birthmo=11 then bbdays=birthdy+305;
  if birthmo=12 then bbdays=birthdy+335;
end;
end;

```

\*\*\*Convert days into week numbers;

```

/* Default age 20 week = 9999 */

```

```

age20wk=9999;

```

```

if age20yr>0 and bbdays>0 then do;
  if age20yr=1999 then do; age20wk=991+ceil((bbdays+5)/7); end;
  if age20yr=2000 then do; age20wk=1043+ceil((bbdays+6)/7); end;
  if age20yr=2001 then do; age20wk=1096+ceil((bbdays+1)/7); end;
  if age20yr=2002 then do; age20wk=1148+ceil((bbdays+2)/7); end;
  if age20yr=2003 then do; age20wk=1200+ceil((bbdays+3)/7); end;
  if age20yr=2004 then do; age20wk=1252+ceil((bbdays+4)/7); end;
end;

```

\*\* Convert Birthdate to week number \*\*

```

/* Default birthdate week=0 if birthdate < 12/30/79 */

```

```

birthwk=0;

```

```

if birthyr>0 and bbdays>0 then do;
  if birthyr=1980 then do; birthwk=ceil((bbdays+2)/7); end;
  if birthyr=1981 then do; birthwk=52+ceil((bbdays+4)/7); end;
  if birthyr=1982 then do; birthwk=104+ceil((bbdays+5)/7); end;
  if birthyr=1983 then do; birthwk=156+ceil((bbdays+6)/7); end;
  if birthyr=1984 then do; birthwk=209+ceil((bbdays)/7); end;
  if birthyr=1985 then do; birthwk=261+ceil((bbdays+2)/7); end;
end;

```

```
** Convert Round 5 Interview Date to week number **;  
***Convert interview month and day to total days (intdays);
```

```
if r5INT_M>0 and r5INT_D>0 then do;  
  if r5INT_M=1 then intdays=r5INT_D;  
  if r5INT_M=2 then intdays=r5INT_D+31;  
  if r5INT_M=3 then intdays=r5INT_D+59;  
  if r5INT_M=4 then intdays=r5INT_D+90;  
  if r5INT_M=5 then intdays=r5INT_D+120;  
  if r5INT_M=6 then intdays=r5INT_D+151;  
  if r5INT_M=7 then intdays=r5INT_D+181;  
  if r5INT_M=8 then intdays=r5INT_D+212;  
  if r5INT_M=9 then intdays=r5INT_D+243;  
  if r5INT_M=10 then intdays=r5INT_D+273;  
  if r5INT_M=11 then intdays=r5INT_D+304;  
  if r5INT_M=12 then intdays=r5INT_D+334;  
end;
```

```
***Convert days into week numbers;
```

```
/* Default interview week = 9999 */
```

```
intwk=9999;
```

```
if r5INT_Y>0 and intdays>0 then do;  
  if r5INT_Y=2001 then do; INTWK=1096+ceil((intdays+1)/7); end;  
    if r5INT_y=2002 then do; intwk=1148+ceil((intdays+2)/7); end;  
  end;
```

```
** Convert Round 4 Interview Date to week number **;
```

```
***Convert interview month and day to total days (r4days);
```

```
if R4INT_M>0 and R4INT_D>0 then do;  
  if R4INT_M=1 then r4days=R4INT_D;  
  if R4INT_M=2 then r4days=R4INT_D+31;  
  if R4INT_M=3 then r4days=R4INT_D+59;  
  if R4INT_M=4 then r4days=R4INT_D+90;  
  if R4INT_M=5 then r4days=R4INT_D+120;  
  if R4INT_M=6 then r4days=R4INT_D+151;  
  if R4INT_M=7 then r4days=R4INT_D+181;  
  if R4INT_M=8 then r4days=R4INT_D+212;  
  if R4INT_M=9 then r4days=R4INT_D+243;  
  if R4INT_M=10 then r4days=R4INT_D+273;  
  if R4INT_M=11 then r4days=R4INT_D+304;  
  if R4INT_M=12 then r4days=R4INT_D+334;  
end;
```

```
***Account for leap year 2000;
```

```
if R4INT_M>0 and R4INT_D>0 and R4INT_Y=2000 then do;  
  if R4INT_M=1 then r4days=R4INT_D;  
  if R4INT_M=2 then r4days=R4INT_D+31;  
  if R4INT_M=3 then r4days=R4INT_D+60;  
  if R4INT_M=4 then r4days=R4INT_D+91;
```

```
if R4INT_M=5 then r4days=R4INT_D+121;
if R4INT_M=6 then r4days=R4INT_D+152;
if R4INT_M=7 then r4days=R4INT_D+182;
if R4INT_M=8 then r4days=R4INT_D+213;
if R4INT_M=9 then r4days=R4INT_D+244;
if R4INT_M=10 then r4days=R4INT_D+274;
if R4INT_M=11 then r4days=R4INT_D+305;
if R4INT_M=12 then r4days=R4INT_D+335;
end;
```

\*\*\*Convert days into week numbers;

/\* Default interview week = 9999 \*/

r4int=9999;

```
if R4INT_Y>0 and r4days>0 then do;
  if R4INT_Y=2000 then do; r4INT=1043+ceil((r4days+6)/7); end;
  if R4INT_y=2001 then do; r4int=1096+ceil((r4days+1)/7); end;
end;
```

\*\* Convert Round 3 Interview Date to week number \*\*;

\*\*\*Convert interview month and day to total days (r3days);

```
if R3INT_M>0 and R3INT_D>0 then do;
  if R3INT_M=1 then r3days=R3INT_D;
  if R3INT_M=2 then r3days=R3INT_D+31;
  if R3INT_M=3 then r3days=R3INT_D+59;
  if R3INT_M=4 then r3days=R3INT_D+90;
  if R3INT_M=5 then r3days=R3INT_D+120;
  if R3INT_M=6 then r3days=R3INT_D+151;
  if R3INT_M=7 then r3days=R3INT_D+181;
  if R3INT_M=8 then r3days=R3INT_D+212;
  if R3INT_M=9 then r3days=R3INT_D+243;
  if R3INT_M=10 then r3days=R3INT_D+273;
  if R3INT_M=11 then r3days=R3INT_D+304;
  if R3INT_M=12 then r3days=R3INT_D+334;
end;
```

\*\*\*Account for leap year 2000;

```
if R3INT_M>0 and R3INT_D>0 and R3INT_Y=2000 then do;
  if R3INT_M=1 then r3days=R3INT_D;
  if R3INT_M=2 then r3days=R3INT_D+31;
  if R3INT_M=3 then r3days=R3INT_D+60;
  if R3INT_M=4 then r3days=R3INT_D+91;
  if R3INT_M=5 then r3days=R3INT_D+121;
  if R3INT_M=6 then r3days=R3INT_D+152;
  if R3INT_M=7 then r3days=R3INT_D+182;
  if R3INT_M=8 then r3days=R3INT_D+213;
  if R3INT_M=9 then r3days=R3INT_D+244;
  if R3INT_M=10 then r3days=R3INT_D+274;
  if R3INT_M=11 then r3days=R3INT_D+305;
  if R3INT_M=12 then r3days=R3INT_D+335;
```

end;

\*\*\*Convert days into week numbers;

/\* Default interview week = 9999 \*/

r3int=9999;

if R3INT\_Y>0 and r3days>0 then do;

  if R3INT\_Y=1999 then do; r3INT=991+ceil((r3days+5)/7); end;

  if R3INT\_y=2000 then do; r3int=1043+ceil((r3days+6)/7); end;

end;

\*\* Convert Date of Round 2 interview to week number \*\*

\*\*\*Convert interview month and day to total days (rd2days);

if R2INT\_M>0 and R2INT\_D>0 then do;

  if R2INT\_M=1 then rd2days=R2INT\_D;

  if R2INT\_M=2 then rd2days=R2INT\_D+31;

  if R2INT\_M=3 then rd2days=R2INT\_D+59;

  if R2INT\_M=4 then rd2days=R2INT\_D+90;

  if R2INT\_M=5 then rd2days=R2INT\_D+120;

  if R2INT\_M=6 then rd2days=R2INT\_D+151;

  if R2INT\_M=7 then rd2days=R2INT\_D+181;

  if R2INT\_M=8 then rd2days=R2INT\_D+212;

  if R2INT\_M=9 then rd2days=R2INT\_D+243;

  if R2INT\_M=10 then rd2days=R2INT\_D+273;

  if R2INT\_M=11 then rd2days=R2INT\_D+304;

  if R2INT\_M=12 then rd2days=R2INT\_D+334;

end;

\*\*\*Convert days into week numbers;

/\* Default interview week = 9999 \*/

R2INT=9999;

if R2INT\_Y>0 and rd2days>0 then do;

  if R2INT\_Y=1998 then do; R2INT=939+ceil((rd2days+4)/7); end;

end;

if R2INT\_Y>0 and rd2days>0 then do;

  if R2INT\_Y=1999 then do; R2INT=991+ceil((rd2days+5)/7); end;

end;

\*\* Convert Date of First Interview (Round 1) to week number \*\*

\*\*\*Convert interview month and day to total days (rd1days);

if R1INT\_M>0 and R1INT\_D>0 then do;

  if R1INT\_M=1 then rd1days=R1INT\_D;

  if R1INT\_M=2 then rd1days=R1INT\_D+31;

  if R1INT\_M=3 then rd1days=R1INT\_D+59;

  if R1INT\_M=4 then rd1days=R1INT\_D+90;

  if R1INT\_M=5 then rd1days=R1INT\_D+120;

  if R1INT\_M=6 then rd1days=R1INT\_D+151;

  if R1INT\_M=7 then rd1days=R1INT\_D+181;

```

if R1INT_M=8 then rd1days=R1INT_D+212;
if R1INT_M=9 then rd1days=R1INT_D+243;
if R1INT_M=10 then rd1days=R1INT_D+273;
if R1INT_M=11 then rd1days=R1INT_D+304;
if R1INT_M=12 then rd1days=R1INT_D+334;
end;

***Convert days into week numbers;
/* Hand edits carried over from Round 2 */
if (pubid= 2 or pubid=5615 or pubid=5902) then do; r1int_y=1997; end;

/* Default interview week = 9999 */
R1INT=9999;

if R1INT_Y>0 and rd1days>0 then do;
  if R1INT_Y=1997 then do; R1INT=887+ceil((rd1days+3)/7); end;
end;
if R1INT_Y>0 and rd1days>0 then do;
  if R1INT_Y=1998 then do; R1INT=939+ceil((rd1days+4)/7); end;
end;

/* Hand edits carried over from Round 2 interview program */
if pubid in (471,476) then do; r2int=990; r2int_y=1998; end;
if pubid=7315 then do; r1int=919; r1int_y=1997; end;

/* Done to create a seamless flow between rounds */
intwk=intwk-1;

/* Round 2 non-interview case */ if r2int_d=-5 then r2int=-5;
/* Round 3 non-interview case */ if r3int_d=-5 then r3int=-5;
/* Round 4 non-interview case */ if r4int_d=-5 then r4int=-5;
/* Round 5 non-interview case */ if e200=-5 then intwk=-5;

/***** Section 3: Convert Start/Stop dates into NLSY97 week numbers *****/
/* This section reads in raw start and stop dates for each job (max=9) and converts them into NLSY97
week numbers. Some start/stop DAYS and MONTHS have been imputed if missing. */
/* ostartm represents the "old" start dates, used when start dates are updated to interview dates. */
array ostartm (i) ostarmo1-ostarmo8;
array ostardt (i) ostardy1-ostardy8;
array ostarty (i) ostaryr1-ostaryr8;
array ostopm (i) ostopmo1-ostopmo8;
array ostodp (i) ostodpy1-ostodpy8;
array ostopy (i) ostopyr1-ostopyr8;
array startm (i) starmo1-starmo8;

array startd (i) stardy1-stardy8;
array starty (i) staryr1-staryr8;
array stopm (i) stopmo1-stopmo8;
array stopd (i) stopdy1-stopdy8;
array stopy (i) stopyr1-stopyr8;

```

```

array UID  (i) UID1-UID8;

array sttdays (i) sttday1-sttday8; /* total days in that year from startdate (to Jan 1) */
array stpdays (i) stpday1-stpday8; /* total days in that year from stopdate (to Jan 1) */
array startwk (i) starw1-starw8;
array stopwk (i) stopw1-stopw8;
array srflag (i) srflg1-srflg8;
array spflag (i) spflg1-spflg8;
array uflag (i) uflag1-uflag8; /* uflag=1 when job startdate is updated */
array smofl (i) smofl1-smofl8; /* dummy equals 1 when a start month is imputed */
array emofl (i) emofl1-emofl8; /* dummy equals 1 when a stop month is imputed */
array self (i) self1-self8;

/* Establish dli interview round */
if r4int_y>0 then do;
  dli_y=r4int_y;
  dli_m=r4int_m;
  dli_d=r4int_d;
  dliwk=r4int;
  r4dli=1;
end;
if r4int_y=-5 and r3int_y>0 then do;
  dli_y=r3int_y;
  dli_m=r3int_m;
  dli_d=r3int_d;
  dliwk=r3int;
  r3dli=1;
end;
if r4int_y=-5 and r3int_y=-5 and r2int_y>0 then do;
  dli_y=r2int_y;
  dli_m=r2int_m;
  dli_d=r2int_d;
  dliwk=r2int;
  r2dli=1;
end;
if r4int_y=-5 and r3int_y=-5 and r2int_y=-5 and r1int_y>0 then do;
  dli_y=r1int_y;
  dli_m=r1int_m;
  dli_d=r1int_d;
  dliwk=r1int;
  r1dli=1;
end;

/* Initialize smofl and emofl */
do i=1 to 8;
  smofl=0; emofl=0;
end;

/* Define old start and stop dates */
do i=1 to 8;
  ostartm=startm;

```

```

ostard=startd;
ostarty=starty;
ostopm=stopm;
ostopd=stopd;
ostopy=stopy;
end;

```

\*\*\*Fill-in start/stop day for those missing;

```

/* flag1 = impute start day (valid month)
   flag2 = impute start month (valid day)
   flag3 = impute start day and month
   flag4 = impute stop day (valid month)
   flag5 = impute stop month (valid day)
   flag6 = impute stop day and month */

```

```

array flag1 (i) flag11-flag18;
array flag2 (i) flag21-flag28;
array flag3 (i) flag31-flag38;
array flag4 (i) flag41-flag48;
array flag5 (i) flag51-flag58;
array flag6 (i) flag61-flag68;

```

```

do i=1 to 8;
  flag1=-4;
  flag2=-4;
  flag3=-4;
  flag4=-4;
  flag5=-4;
  flag6=-4;
end;

```

/\* We are now inputting missing start days to 1, missing stop days to 28, missing start months to 1 (Jan.) and missing stop months to 12 (Dec.). \*/

/\* Reset flag(1-3) to zero when the start year is valid \*/

```

do over starty;
  if starty>0 then do;
    flag1=0; flag2=0; flag3=0;
    if startm>0 and startd<=0 then do;
      startd=1;
      flag1=1;
      srflag=1;
    end;
    if startm<=0 and startd>0 then do;
      startm=1;
      flag2=1;
      srflag=1;
      smofl=1;
    end;
    if startm<=0 and startd<=0 then do;
      startm=1;

```

```

startd=1;
flag3=1;
srflag=1;
smofl=1;
end;
end;
end;

do over stopy;
  if stopy>0 then do;
    flag4=0; flag5=0; flag6=0;
    if stopm>0 and stopd<=0 then do;
      stopd=28;
      flag4=1;
      spflag=1;
    end;
    if stopm<=0 and stopd>0 then do;
      stopm=12;
      flag5=1;
      spflag=1;
      emofl=1;
    end;
    if stopm<=0 and stopd<=0 then do;
      stopm=12;
      stopd=28;
      flag6=1;
      spflag=1;
      emofl=1;
    end;
  /* The following lines prevent imputed values for stop months and days to exceed the Round 5 interview date. */
  if stopy=r5int_y and stopm=r5int_m and stopd>r5int_d and spflag=1 then do;
    stopd=r5int_d;
  end;
  if stopy=r5int_y and stopm>r5int_m then do;
    stopm=r5int_m;
    stopd=r5int_d;
  end;
  end;
end;

array backr (i) backr1-backr9;
do i=1 to 8;
  backr=0;
end;

/* This respondent reports a previous job that begins and ends before the dli. For the Round 5 c.v.'s, this job is irrelevant. */
if pubid=3923 then do;
  stardy1=-4; starmo1=-4; staryr1=-4; uid1=-4;
  stopdy1=-4; stopmo1=-4; stopyr1=-4;

```

```
end;
```

```
/* The following lines of code account for the cases where the respondent has a start date for a job reported in an earlier round that comes before the dli interview date. The idea is to only count the weeks employed from dli to today. Then, the activity from previous rounds will be added to the Round 5 activity to get the full history. To achieve this, all jobs reported in earlier rounds (UID's begin with 97,98,1999, or 2000) will have their startdates updated to the dli interview date. The start week and stop week for these jobs will be counted the same. */
```

```
/* Changed "uid>0" condition to "uid>-5," since there is one respondent with UID=-3. This has not been programmed in previous rounds since no UID's in previous rounds equaled -2 or -3. */
```

```
do i=1 to 8;
```

```
  uflag=0;          /* Initialize uflag */
  if -5<uid<200100 and startd>-4 and backr=0 then do;
    if starty=dli_y and startm=dli_m and startd<dli_d and starty>0 then do;
      startd=dli_d;
      uflag=1;
    end;
    if starty=dli_y and startm<=dli_m and starty>0 then do;
      startm=dli_m;
      startd=dli_d;
      uflag=1;
    end;
    if starty<=dli_y and starty>0 then do;
      starty=dli_y;
      startm=dli_m;
      startd=dli_d;
      uflag=1;
    end;
  end;
end;
```

```
***Convert START month and day to total days;
```

```
do over startm;
```

```
  if startm>0 and startd>0 then do;
    if startm=1 then sttdays=startd;
    if startm=2 then sttdays=startd+31;
    if startm=3 then sttdays=startd+59;
    if startm=4 then sttdays=startd+90;
    if startm=5 then sttdays=startd+120;
    if startm=6 then sttdays=startd+151;
    if startm=7 then sttdays=startd+181;
    if startm=8 then sttdays=startd+212;
    if startm=9 then sttdays=startd+243;
    if startm=10 then sttdays=startd+273;
    if startm=11 then sttdays=startd+304;
    if startm=12 then sttdays=startd+334;
  end;
end;
```

\*\*\*Account for leap years;

```
do over starty;
if starty=1980 or starty=1984 or starty=1988 or starty=1992 or starty=1996 or starty=2000 then do;
  if startm>0 and startd>0 then do;
    if startm=1 then sttdays=startd;
    if startm=2 then sttdays=startd+31;
    if startm=3 then sttdays=startd+60;
    if startm=4 then sttdays=startd+91;
    if startm=5 then sttdays=startd+121;
    if startm=6 then sttdays=startd+152;
    if startm=7 then sttdays=startd+182;
    if startm=8 then sttdays=startd+213;
    if startm=9 then sttdays=startd+244;
    if startm=10 then sttdays=startd+274;
    if startm=11 then sttdays=startd+305;
    if startm=12 then sttdays=startd+335;
  end;
end;
end;
```

\*\*\*Convert STOP month and day to total days;

```
do over stopm;
if stopm>0 and stopd>0 then do;
  if stopm=1 then stpdays=stopd;
  if stopm=2 then stpdays=stopd+31;
  if stopm=3 then stpdays=stopd+59;
  if stopm=4 then stpdays=stopd+90;
  if stopm=5 then stpdays=stopd+120;
  if stopm=6 then stpdays=stopd+151;
  if stopm=7 then stpdays=stopd+181;
  if stopm=8 then stpdays=stopd+212;
  if stopm=9 then stpdays=stopd+243;
  if stopm=10 then stpdays=stopd+273;
  if stopm=11 then stpdays=stopd+304;
  if stopm=12 then stpdays=stopd+334;
end;
end;
```

\*\*\*Account for leap years;

```
do over stopy;
if stopy=1980 or stopy=1984 or stopy=1988 or stopy=1992 or stopy=1996 or stopy=2000 then do;
  if stopm>0 and stopd>0 then do;
    if stopm=1 then stpdays=stopd;
    if stopm=2 then stpdays=stopd+31;
    if stopm=3 then stpdays=stopd+60;
    if stopm=4 then stpdays=stopd+91;
    if stopm=5 then stpdays=stopd+121;
    if stopm=6 then stpdays=stopd+152;
```

```

if stopm=7 then stpdays=stopd+182;
if stopm=8 then stpdays=stopd+213;
if stopm=9 then stpdays=stopd+244;
if stopm=10 then stpdays=stopd+274;
if stopm=11 then stpdays=stopd+305;
if stopm=12 then stpdays=stopd+335;
end;
end;
end;

***Convert days into week numbers;

/* Create year flag variable */
array byear (i) byear1-byear9;
array eyear (i) eyear1-eyear9;

do over starty;
  if starty>0 and sttdays>0 then do;
    if starty=1980 then do; startwk=ceil((sttdays+2)/7); end;
    if starty=1981 then do; startwk=52+ceil((sttdays+4)/7); end;
    if starty=1982 then do; startwk=104+ceil((sttdays+5)/7); end;
    if starty=1983 then do; startwk=156+ceil((sttdays+6)/7); end;
    if starty=1984 then do; startwk=209+ceil((sttdays)/7); end;
    if starty=1985 then do; startwk=261+ceil((sttdays+2)/7); end;
    if starty=1986 then do; startwk=313+ceil((sttdays+3)/7); end;
    if starty=1987 then do; startwk=365+ceil((sttdays+4)/7); end;
    if starty=1988 then do; startwk=417+ceil((sttdays+5)/7); end;
    if starty=1989 then do; startwk=470+ceil((sttdays)/7); end;
    if starty=1990 then do; startwk=522+ceil((sttdays+1)/7); end;
    if starty=1991 then do; startwk=574+ceil((sttdays+2)/7); end;
    if starty=1992 then do; startwk=626+ceil((sttdays+3)/7); end;
    if starty=1993 then do; startwk=678+ceil((sttdays+5)/7); end;
    if starty=1994 then do; startwk=730+ceil((sttdays+6)/7); end;
    if starty=1995 then do; startwk=783+ceil((sttdays)/7); end;
    if starty=1996 then do; startwk=835+ceil((sttdays+1)/7); end;
    if starty=1997 then do; startwk=887+ceil((sttdays+3)/7); end;
    if starty=1998 then do; startwk=939+ceil((sttdays+4)/7); end;
    if starty=1999 then do; startwk=991+ceil((sttdays+5)/7); end;
    if starty=2000 then do; startwk=1043+ceil((sttdays+6)/7); end;
    if starty=2001 then do; startwk=1096+ceil((sttdays+1)/7); end;
    if starty=2002 then do; startwk=1148+ceil((sttdays+2)/7); end;
  end;
  if starty<0 and starty>-4 then do;
    startwk=-3;
  end;
  if -4<ostarty<0 then byear=1;
end;

do over stopy;
  if stopy>0 and stpdays>0 then do;
    if stopy=1980 then do; stopwk=ceil((stpdays+2)/7); end;

```

```

if stopy=1981 then do; stopwk=52+ceil((stpdays+4)/7); end;
if stopy=1982 then do; stopwk=104+ceil((stpdays+5)/7); end;
if stopy=1983 then do; stopwk=156+ceil((stpdays+6)/7); end;
if stopy=1984 then do; stopwk=209+ceil((stpdays)/7); end;
if stopy=1985 then do; stopwk=261+ceil((stpdays+2)/7); end;
if stopy=1986 then do; stopwk=313+ceil((stpdays+3)/7); end;
if stopy=1987 then do; stopwk=365+ceil((stpdays+4)/7); end;
if stopy=1988 then do; stopwk=417+ceil((stpdays+5)/7); end;
if stopy=1989 then do; stopwk=470+ceil((stpdays)/7); end;
if stopy=1990 then do; stopwk=522+ceil((stpdays+1)/7); end;
if stopy=1991 then do; stopwk=574+ceil((stpdays+2)/7); end;
if stopy=1992 then do; stopwk=626+ceil((stpdays+3)/7); end;
if stopy=1993 then do; stopwk=678+ceil((stpdays+5)/7); end;
if stopy=1994 then do; stopwk=730+ceil((stpdays+6)/7); end;
if stopy=1995 then do; stopwk=783+ceil((stpdays)/7); end;
if stopy=1996 then do; stopwk=835+ceil((stpdays+1)/7); end;
if stopy=1997 then do; stopwk=887+ceil((stpdays+3)/7); end;
if stopy=1998 then do; stopwk=939+ceil((stpdays+4)/7); end;
if stopy=1999 then do; stopwk=991+ceil((stpdays+5)/7); end;
if stopy=2000 then do; stopwk=1043+ceil((stpdays+6)/7); end;
if stopy=2001 then do; stopwk=1096+ceil((stpdays+1)/7); end;
if stopy=2002 then do; stopwk=1148+ceil((stpdays+2)/7); end;
end;
if stopy<0 and stopy>-4 then do;
    stopwk=-3;
end;
if -4<ostarty<0 then eyear=1;
end;

```

/\* To create a seamless list of information from dli Round to Round 5, the following lines of code are included. By decreasing the Round 5 interview date by one, (which was done in emp\_interview\_r5.sas) jobs that are worked up to the interview date are also decreased by one. This is done so that the Round 5 interview week will not be counted twice for respondents with a job during that time. The same procedure is used in the previous rounds. \*/

```

do over stopwk;
if stopwk>0 and UID>0 then do;
    if stopwk>intwk then do;
        stopwk=intwk;
    end;
end;
end;

```

/\* The following lines considers jobs that begin the same week as the Round 5 interview date. Since we are updating the Round 5 interview week by -1, we need to account for jobs that start in the same week or tenures of -1 will result. \*/

```

do over startwk;
if startwk>0 and UID>0 then do;
    if startwk>intwk then do;
        startwk=intwk;
    end;
end;

```

```

    end;
end;
end;

/* The job start date will be cut off at the week of Jan.1 of the respondent's 18th birthday for self-employed jobs. This is done so every starts the self-employed jobs at the same point. The 18th birthday week was not chosen because too many jobs were completely eliminated using the actual 18th birthday. This method cuts off 2 jobs entirely (both the start and stop week are before Jan. 1 of the 18th birthday), and shortens 26 other jobs.*/
array bself (i) bself1-bself9;
array eself (i) eself1-eself9;
do over startwk;
  if self=1 and 0<startwk<=(ag18jan-1) then do;
    bself=1;
    bcheck=1;
  end;
  if self=1 and 0<startwk<=(ag18jan-1) and 0<stopwk<=(ag18jan-1) then do;
    eself=1;
    echeck=1;
  end;
end;
do over startwk;
  if bself=1 then do;
    startwk=ag18jan;
  end;
  if eself=1 then do;
    startwk=.;
    stopwk=.;
  end;
end;

/* To account for Round 4 non-interview cases */
if starmo1=-5 then do;
  do over startwk;
    startwk=-5;
    stopwk=-5;
  end;
end;

/* Correcting for imputed values that resulted in the start date being later than the stop date. In these cases, the imputed date will be updated to the good date.*/
do i=1 to 8;
  if startwk>stopwk and (flag1=1 or flag2=1 or flag3=1) then do;
    startwk=stopwk;
  end;
  if startwk>stopwk and (flag4=1 or flag5=1 or flag6=1) then do;
    stopwk=startwk;
  end;
end;

```

```

/* Check for back-reporters, i.e. people that report a new job (UID>200000) that begins before the last
interview date. */
do over UID;
  if startwk<r4int and startwk>0 and r4int>0 then do;
    back4=1;
    back=1;
  end;
  if startwk<r3int and startwk>0 and r4int=-5 and r3int>0 then do;
    back3=1;
    back=1;
  end;
  if startwk<r2int and startwk>0 and r4int=-5 and r2int>0 and r3int=-5 then do;
    back2=1;
    back=1;
  end;
  if startwk<r1int and startwk>0 and r4int=-5 and r1int>0 and r2int=-5 and r3int=-5 then do;
    back1=1;
    back=1;
  end;
end;

array br (i) br1-br9;

do over UID;
  if startwk<dliwk and startwk>0 and dliwk>0 then do;
    back=1;
    br=1;
  end;
end;

do over UID;
  if startwk<r4int and self=1 and startwk>0 and r4int>0 then do;
    back4=1;
    br=1;
  end;
  if startwk<r3int and self=1 and startwk>0 and r4int=-5 and r3int>0 then do;
    back3=1;
    br=1;
  end;
  if startwk<r2int and self=1 and startwk>0 and r4int=-5 and r3int=-5 then do;
    back2=1;
    br=1;
  end;
  if startwk<r1int and self=1 and startwk>0 and r4int=-5 and r2int=-5 and r3int=-5 then do;
    back1=1;
    br=1;
  end;
end;

```

/\* The following 3 people have bad month startdates and new job UID's (e.g., 200101). The imputations set these bad start months to January, which predates their dli. \*/

```
if pubid=1648 then starw2=1050;  
if pubid=8366 then starw2=1047;  
if pubid=8696 then starw2=1039;
```

\*\*\*\*\* Section 4: Convert dates of within-job gaps into NLSY97 week numbers.

\*\*\*\*\*

\*\* JOB 1 GAPS \*\*;

/\* These variables are read as follows:

BGDY1\_1 = Begin day of within-job gap 1 on job 1  
EGMO1\_5 = End month of within-job gap 5 on job 1  
BGAP1\_3 = Begin week of within-job gap 3 on job 1 [CREATED] \*/

```
array bgdy (i) BGDY1_1-BGYD1_6;  
array bgmo (i) BGMO1_1-BGMO1_6;  
array bgyr (i) BGYR1_1-BGYR1_6;  
array egdy (i) EGDY1_1-EGDY1_6;  
array egmo (i) EGMO1_1-EGMO1_6;  
array egyr (i) EGYR1_1-EGYR1_6;
```

```
array bdays (i) bday1_1-bday1_6;  
array edays (i) eday1_1-eday1_6;  
array bweek (i) bgap1_1-bgap1_6;  
array eweek (i) egap1_1-egap1_6;  
array bflag (i) bflg1_1-bflg1_6;  
array eflag (i) eflg1_1-eflg1_6;
```

```
array bgfl (i) bgfl1_1-bgfl1_6;  
array egfl (i) egfl1_1-egfl1_6;
```

/\* ONLY IMPUTE START/STOP DATES IF DAY IS MISSING \*/

/\* Fill-in start day for those missing \*/

/\* Missing gap days (start and stop) are now imputed to 1. Repeated without comment at each job. \*/  
do over bgyr;

```
if bgyr>0 then do;  
  if bgmo>0 and bgdy<=0 then do;  
    bgdy=1;  
    bflag=1;  
  end;  
  if bgyr=staryr1 and bgmo=starmo1 and bgdy<stardy1 and bflag=1 then do; /* This accounts for  
beginning gap dates before job */  
    bgdy=stardy1; /* start date. */  
  end;  
end;  
end;
```

/\* Fill-in stop day for those missing \*/  
do over egyr;

```
if egyr>0 then do;  
  if egmo>0 and egdy<=0 then do;  
    egdy=1;
```

```

        eflag=1;
        end;
/* This accounts for end gap dates after job start date */
        if egyr=stopyr1 and egmo=stopmo1 and egdy>stopdy1 and eflag=1 then do;
            egdy=stopdy1;
        end;
        end;
        end;

/*Set flag for gap exists but invalid data*/
do over bgyr;
    bgfl=-4;
    if (-4 < bgmo < 0) or (-4 < bgyr <0) then do;
        bgfl=1;
    end;
end;
end;

do over egyr;
    egfl=-4;
    if (-4 < egyr < 0) or (-4 < egmo < 0) then do;
        egfl=1;
    end;
end;
end;

***Identify within-job gaps on JOB 1 ***;
***Convert gap dates to week numbers ***;
***Convert START month and day to total days (BDAYS);

do over bgmo;
    if bgmo>0 and bgdy>0 then do;
        if bgmo=1 then bdays=bgdy;
        if bgmo=2 then bdays=bgdy+31;
        if bgmo=3 then bdays=bgdy+59;
        if bgmo=4 then bdays=bgdy+90;
        if bgmo=5 then bdays=bgdy+120;
        if bgmo=6 then bdays=bgdy+151;
        if bgmo=7 then bdays=bgdy+181;
        if bgmo=8 then bdays=bgdy+212;
        if bgmo=9 then bdays=bgdy+243;
        if bgmo=10 then bdays=bgdy+273;
        if bgmo=11 then bdays=bgdy+304;
        if bgmo=12 then bdays=bgdy+334;
    end;
end;

***Account for leap years;
do over bgyr;
    if bgyr=1980 or bgyr=1984 or bgyr=1988 or bgyr=1992 or bgyr=1996 or bgyr=2000 then do;
        if bgmo>0 and bgdy>0 then do;
            if bgmo=1 then bdays=bgdy;
            if bgmo=2 then bdays=bgdy+31;

```

```

if bgmo=3 then bdays=bgdy+60;
if bgmo=4 then bdays=bgdy+91;
if bgmo=5 then bdays=bgdy+121;
if bgmo=6 then bdays=bgdy+152;
if bgmo=7 then bdays=bgdy+182;
if bgmo=8 then bdays=bgdy+213;
if bgmo=9 then bdays=bgdy+244;
if bgmo=10 then bdays=bgdy+274;
if bgmo=11 then bdays=bgdy+305;
if bgmo=12 then bdays=bgdy+335;
end;
end;
end;

```

\*\*\*Convert STOP month and day to total days (EDAYS);  
do over egmo;

```

if egmo>0 and egdy>0 then do;
  if egmo=1 then edays=egdy;
  if egmo=2 then edays=egdy+31;
  if egmo=3 then edays=egdy+59;
  if egmo=4 then edays=egdy+90;
  if egmo=5 then edays=egdy+120;
  if egmo=6 then edays=egdy+151;
  if egmo=7 then edays=egdy+181;
  if egmo=8 then edays=egdy+212;
  if egmo=9 then edays=egdy+243;
  if egmo=10 then edays=egdy+273;
  if egmo=11 then edays=egdy+304;
  if egmo=12 then edays=egdy+334;
end;
end;

```

\*\*\*Account for leap years;

do over egyr;

```

if egyr=1980 or egyr=1984 or egyr=1988 or egyr=1992 or egyr=1996 or egyr=2000 then do;
  if egmo>0 and egdy>0 then do;
    if egmo=1 then edays=egdy;
    if egmo=2 then edays=egdy+31;
    if egmo=3 then edays=egdy+60;
    if egmo=4 then edays=egdy+91;
    if egmo=5 then edays=egdy+121;
    if egmo=6 then edays=egdy+152;
    if egmo=7 then edays=egdy+182;
    if egmo=8 then edays=egdy+213;
    if egmo=9 then edays=egdy+244;
    if egmo=10 then edays=egdy+274;
    if egmo=11 then edays=egdy+305;
    if egmo=12 then edays=egdy+335;
  end;
end;
end;

```

\*\*\*Convert days into week numbers;  
 \*\* Note: Use this program takes the week following the actual start of the gap    \*\*  
 \*\*        as the measure of when the non-working period begins.        \*\*;

do over bgyr;

```

if bgyr>0 and bdays>0 then do;
  if bgyr=1980 then do; bweek=ceil((bdays+2)/7); end;
  if bgyr=1981 then do; bweek=52+ceil((bdays+4)/7); end;
  if bgyr=1982 then do; bweek=104+ceil((bdays+5)/7); end;
  if bgyr=1983 then do; bweek=156+ceil((bdays+6)/7); end;
  if bgyr=1984 then do; bweek=209+ceil((bdays)/7); end;
  if bgyr=1985 then do; bweek=261+ceil((bdays+2)/7); end;
  if bgyr=1986 then do; bweek=313+ceil((bdays+3)/7); end;
  if bgyr=1987 then do; bweek=365+ceil((bdays+4)/7); end;
  if bgyr=1988 then do; bweek=417+ceil((bdays+5)/7); end;
  if bgyr=1989 then do; bweek=470+ceil((bdays)/7); end;
  if bgyr=1990 then do; bweek=522+ceil((bdays+1)/7); end;
  if bgyr=1991 then do; bweek=574+ceil((bdays+2)/7); end;
  if bgyr=1992 then do; bweek=626+ceil((bdays+3)/7); end;
  if bgyr=1993 then do; bweek=678+ceil((bdays+5)/7); end;
  if bgyr=1994 then do; bweek=730+ceil((bdays+6)/7); end;
  if bgyr=1995 then do; bweek=783+ceil((bdays)/7); end;
  if bgyr=1996 then do; bweek=835+ceil((bdays+1)/7); end;
  if bgyr=1997 then do; bweek=887+ceil((bdays+3)/7); end;
  if bgyr=1998 then do; bweek=939+ceil((bdays+4)/7); end;
  if bgyr=1999 then do; bweek=991+ceil((bdays+5)/7); end;
  if bgyr=2000 then do; bweek=1043+ceil((bdays+6)/7); end;
  if bgyr=2001 then do; bweek=1096+ceil((bdays+1)/7); end;
  if bgyr=2002 then do; bweek=1148+ceil((bdays+2)/7); end;
  if bweek>0 then do;
    bweek=bweek+1;
  end;
end;
end;
```

do over egyr;

```

if egyr>0 and edays>0 then do;
  if egyr=1980 then do; eweek=ceil((edays+2)/7); end;
  if egyr=1981 then do; eweek=52+ceil((edays+4)/7); end;
  if egyr=1982 then do; eweek=104+ceil((edays+5)/7); end;
  if egyr=1983 then do; eweek=156+ceil((edays+6)/7); end;
  if egyr=1984 then do; eweek=209+ceil((edays)/7); end;
  if egyr=1985 then do; eweek=261+ceil((edays+2)/7); end;
  if egyr=1986 then do; eweek=313+ceil((edays+3)/7); end;
  if egyr=1987 then do; eweek=365+ceil((edays+4)/7); end;
  if egyr=1988 then do; eweek=417+ceil((edays+5)/7); end;
  if egyr=1989 then do; eweek=470+ceil((edays)/7); end;
  if egyr=1990 then do; eweek=522+ceil((edays+1)/7); end;
  if egyr=1991 then do; eweek=574+ceil((edays+2)/7); end;
  if egyr=1992 then do; eweek=626+ceil((edays+3)/7); end;
```

```

if egyr=1993 then do; eweek=678+ceil((edays+5)/7); end;
if egyr=1994 then do; eweek=730+ceil((edays+6)/7); end;
if egyr=1995 then do; eweek=783+ceil((edays)/7); end;
if egyr=1996 then do; eweek=835+ceil((edays+1)/7); end;
if egyr=1997 then do; eweek=887+ceil((edays+3)/7); end;
if egyr=1998 then do; eweek=939+ceil((edays+4)/7); end;
if egyr=1999 then do; eweek=991+ceil((edays+5)/7); end;
if egyr=2000 then do; eweek=1043+ceil((edays+6)/7); end;
if egyr=2001 then do; eweek=1096+ceil((edays+1)/7); end;
if egyr=2002 then do; eweek=1148+ceil((edays+2)/7); end;
if eweek>0 then do;
    eweek=eweek-1;
end;
end;
array update (i) update1-update10;

/* The following lines omit gap start and stop dates for gaps less than one work week (5 days) */
do over bdays;
if edays-bdays<5 and bweek>eweek and bdays ne . and edays ne . then do;
    bweek=.;
    eweek=.;
end;
end;

/* The following lines are for end gap dates that exceed the Round4 interview date due to our rounding
methods. */
do over eweek;
if 0<eweek<1400 and eweek>intwk then do;
    eweek=intwk;
end;
end;

/* The following omits cases where bweek>eweek, which are caused when missing values are
substituted in. This can cause problems when writing programs for the created variables. This situation
will be fixed by making eweek and bweek the same. This is repeated (without comment) for each job. */

do over bweek;
if 0<eweek<1200 then do;
    if bweek>eweek then do;
        bweek=eweek;
        update=1;
    end;
end;
end;

/* If a respondent reports a gap on previous round's job that occurred before the dli week, then we will
not use this information. However, a newly reported job (i.e., 2002 UID) containing within job gaps
that occurred before the dli week will be counted. This is repeated (without comment) for each job. Note
that we have respondents who were not interviewed in previous round(s). The same process is done for

```

```

these people, except we use the date of last interview. */
do over bweek;
  if UID1>0 and dliwk>0 and bweek>0 and eweek>0 and backr1 ne 1 then do;
    if dliwk>bweek and dliwk>eweek then do; /* 11-27-00 Entire gap is before dli. */
      bweek=.;
      eweek=.;
      end;
    if dliwk<eweek and dliwk>bweek then do; /* 11-27-00 B.gap is before dli, e.gap is after dli. */
      bweek=dliwk;
      end;
    end;
  end;

/* The following erases gaps that occur after the job has ended, only in cases where there are no imputed
job start or stop months or years (imputed job start/stop days acceptable) */
do over bweek;
  if starw1>0 and stopw1>0 and smof1l=0 and emof1l=0 then do;
    if bweek>stopw1 then do;
      bweek=.;
      end;
    if eweek>stopw1 then do;
      eweek=.;
      end;
    end;
  end;

/* To correct for bad gap information.*/
do over bdays;
  if eweek>stopw1 and eweek ne . and stopw1>0 then eweek=stopw1;
  if bweek<starw1 and bweek ne . and bweek>0 then bweek=starw1;
  if bweek>stopw1 and bweek ne . and stopw1>0 then huh=1;
  if eweek<starw1 and eweek ne . and starw1>0 then huh=1;
end;

/*At this point the program loops through the same code for gaps on jobs 2-5. There are no gaps
reported by any respondent for jobs 6-9. Due to space considerations, this code is not repeated here.
For more information please contact NLS User Services.*/

***** Section 5: Weeks worked by respondent on each job *****/
/* This program counts the weeks worked by the respondent and removes the within job gaps. It will
place a "1" into weeks where the respondent was employed. Weeks range from the first week of 1980 to
the last week of 2002, for a total of 1200 weeks. */

/* Variable definitions of problem cases are (with frequencies in parenthesis)
back1: backreporter, job 1
back2: backreporter, job 2
...
back5: backreporter, job 5
Note that the only backreporting allowed is when the a 2002 job begins and ends before the date of last
interview.

```

They are hand edits at the "total" programs. \*/

```
array job1wks (i) wk1_1-wk1_1200;
array job2wks (i) wk2_1-wk2_1200;
array job3wks (i) wk3_1-wk3_1200;
array job4wks (i) wk4_1-wk4_1200;
array job5wks (i) wk5_1-wk5_1200;
array job6wks (i) wk6_1-wk6_1200;
array job7wks (i) wk7_1-wk7_1200;
array job8wks (i) wk8_1-wk8_1200;
array starw (i) starw1-starw9;
array stopw (i) stopw1-stopw9;
```

\* Default Settings;

```
do i=1 to 1200;
  job1wks=0;
  job2wks=0;
  job3wks=0;
  job4wks=0;
  job5wks=0;
  job6wks=0;
  job7wks=0;
  job8wks=0;
  job9wks=0;
end;
```

\* Change military start and stop dates to missing since only civilian jobs are counted;

```
if milflag1=1 then do; starw1=.; stopw1=.; end;
if milflag2=1 then do; starw2=.; stopw2=.; end;
if milflag3=1 then do; starw3=.; stopw3=.; end;
if milflag4=1 then do; starw4=.; stopw4=.; end;
if milflag5=1 then do; starw5=.; stopw5=.; end;
if milflag6=1 then do; starw6=.; stopw6=.; end;
if milflag7=1 then do; starw7=.; stopw7=.; end;
if milflag8=1 then do; starw8=.; stopw8=.; end;
if milflag9=1 then do; starw9=.; stopw9=.; end;
```

/\* Define rd5wk as the maximum of dliwk and age14wk. This is used for bad start/stop weeks. \*/

```
if dliwk>0 then do;
  if dliwk>age14wk then do;
    rd5wk=dliwk;
  end;
  if age14wk=>dliwk then do;
    rd5wk=age14wk;
  end;
end;
```

\*\*\* TOTAL WEEKS WORKED ON JOB 1 \*\*\*

```
starfl_1=0;
stopfl_1=0;
```

```

if starw1=-3 and uid1 ne -5 then do; starw1=rd5wk; starfl_1=1; end;
if stopw1=-3 and uid1 ne -5 then do; stopw1=intwk; stopfl_1=1; end;
if smofl1=1 then do; starfl_1=1; end;
if emofl1=1 then do; stopfl_1=1; end;

if starw1>0 and stopw1>0 then do; /* [1] */
  do i=(starw1) to (stopw1); job1wks=1; end;

  *** Remove gap 1 on job 1;
  if bgap1_1>0 & egap1_1>0 then do;
    do i=(bgap1_1) to (egap1_1); job1wks=0; end;
  end;
  *** Remove gap 2 on job 1;
  if bgap1_2>0 & egap1_2>0 then do;
    do i=(bgap1_2) to (egap1_2); job1wks=0; end;
  end;
  *** Remove gap 3 on job 1;
  if bgap1_3>0 & egap1_3>0 then do;
    do i=(bgap1_3) to (egap1_3); job1wks=0; end;
  end;
  *** Remove gap 4 on job 1;
  if bgap1_4>0 & egap1_4>0 then do;
    do i=(bgap1_4) to (egap1_4); job1wks=0; end;
  end;
  *** Remove gap 5 on job 1;
  if bgap1_5>0 & egap1_5>0 then do;
    do i=(bgap1_5) to (egap1_5); job1wks=0; end;
  end;
  *** Remove gap 6 on job 1;
  if bgap1_6>0 & egap1_6>0 then do;
    do i=(bgap1_6) to (egap1_6); job1wks=0; end;
  end;

  *** Remove gap 1 on job 1 - beginning gap date bad;
  if bgfl1_1=1 & egap1_1>0 then do;
    do i=(starw1) to (egap1_1);
      job1wks=-3;
      gpfl1_1=1;
    end;
  end;
  *** Remove gap 1 on job 1 - end gap date bad;
  if bgap1_1>0 & egfl1_1=1 then do;
    do i=(bgap1_1) to min(stopw1,bgap1_2-1,bgap1_3-1,bgap1_4-1,bgap1_5-1,bgap1_6-1,
      egap1_2+1,egap1_3+1,egap1_4+1,egap1_5+1,egap1_6+1);
      job1wks=-3;
      gpfl1_1=1;
    end;
  end;
  *** Remove gap 1 on job 1 - both gap dates bad;
  if bgfl1_1=1 & egfl1_1=1 then do;
    do i=(starw1) to min(stopw1,bgap1_2-1,bgap1_3-1,bgap1_4-1,bgap1_5-1,bgap1_6-1,

```

```

egap1_2+1,egap1_3+1,egap1_4+1,egap1_5+1,egap1_6+1);
  job1wks=-3;
  gpfl1_1=1;
end;
end;

*** Remove gap 2 on job 1 - beginning gap date bad;
if bgfl1_2=1 & egap1_2>0 then do;
  do i=max(starw1,bgap1_1-1,egap1_1+1) to (egap1_2);
    job1wks=-3;
    gpfl1_2=1;
  end;
end;
*** Remove gap 2 on job 1 - end gap date bad;
if bgap1_2>0 & egfl1_2=1 then do;
  do i=(bgap1_2) to min(stopw1,bgap1_3-1,bgap1_4-1,bgap1_5-1,bgap1_6-1,
    egap1_3+1,egap1_4+1,egap1_5+1,egap1_6+1);
    job1wks=-3;
    gpfl1_2=1;
  end;
end;
*** Remove gap 2 on job 1 - both gap dates bad;
if bgfl1_2=1 & egfl1_2=1 then do;
  do i=max(starw1,bgap1_1-1,egap1_1+1) to min(stopw1,bgap1_3-1,bgap1_4-1,bgap1_5-1,bgap1_6-
1,
  egap1_3+1,egap1_4+1,egap1_5+1,egap1_6+1);
    job1wks=-3;
    gpfl1_2=1;
  end;
end;

*** Remove gap 3 on job 1 - beginning gap date bad;
if bgfl1_3=1 & egap1_3>0 then do;
  do i=max(starw1,bgap1_1-1,bgap1_2-1,egap1_1+1,egap1_2+1) to (egap1_3);
    job1wks=-3;
    gpfl1_3=1;
  end;
end;
*** Remove gap 3 on job 1 - end gap date bad;
if bgap1_3>0 & egfl1_3=1 then do;
  do i=(bgap1_3) to min(stopw1,bgap1_4-1,bgap1_5-1,bgap1_6-1,
    egap1_4+1,egap1_5+1,egap1_6+1);
    job1wks=-3;
    gpfl1_3=1;
  end;
end;
*** Remove gap 3 on job 1 - both gap dates bad;
if bgfl1_3=1 & egfl1_3=1 then do;
  do i=max(starw1,bgap1_1-1,bgap1_2-1,egap1_1+1,egap1_2+1) to min(stopw1,bgap1_4-1,bgap1_5-
1,bgap1_6-1,
  egap1_4+1,egap1_5+1,egap1_6+1);

```

```

job1wks=-3;
gpfl1_3=1;
end;
end;

*** Remove gap 4 on job 1 - beginning gap date bad;
if bgfl1_4=1 & egap1_4>0 then do;
  do i=max(starw1,bgap1_1-1,bgap1_2-1,bgap1_3-1,egap1_1+1,egap1_2+1,egap1_3+1) to (egap1_4);
    job1wks=-3;
    gpfl1_4=1;
  end;
end;
*** Remove gap 4 on job 1 - end gap date bad;
if bgap1_4>0 & egfl1_4=1 then do;
  do i=(bgap1_4) to min(stopw1,bgap1_5-1,egap1_5+1,egap1_6+1);
    job1wks=-3;
    gpfl1_4=1;
  end;
end;
*** Remove gap 4 on job 1 - both gap dates bad;
if bgfl1_4=1 & egfl1_4=1 then do;
  do i=max(starw1,bgap1_1-1,bgap1_2-1,bgap1_3-1,egap1_1+1,egap1_2+1,egap1_3+1) to min
(stopw1,bgap1_5-1,bgap1_6-1,
  egap1_5+1,egap1_6+1);
    job1wks=-3;
    gpfl1_4=1;
  end;
end;

*** Remove gap 5 on job 1 - beginning gap date bad;
if bgfl1_5=1 & egap1_5>0 then do;
  do i=max(starw1,bgap1_1-1,bgap1_2-1,bgap1_3-1,bgap1_4-
1,egap1_1+1,egap1_2+1,egap1_3+1,egap1_4+1) to (egap1_5);
    job1wks=-3;
    gpfl1_5=1;
  end;
end;
*** Remove gap 5 on job 1 - end gap date bad;
if bgap1_5>0 & egfl1_5=1 then do;
  do i=(bgap1_5) to min(stopw1,bgap1_6-1,egap1_6+1);
    job1wks=-3;
    gpfl1_5=1;
  end;
end;
*** Remove gap 5 on job 1 - both gap dates bad;
if bgfl1_5=1 & egfl1_5=1 then do;
  do i=max(starw1,bgap1_1-1,bgap1_2-1,bgap1_3-1,bgap1_4-
1,egap1_1+1,egap1_2+1,egap1_3+1,egap1_4+1)
  to min(stopw1,bgap1_6-1,egap1_6+1);
    job1wks=-3;
    gpfl1_5=1;
  end;

```

```

    end;
end;

*** Remove gap 6 on job 1 - beginning gap date bad;
if bgfl1_6=1 & egap1_6>0 then do;
  do i=max(starw1,bgap1_1-1,bgap1_2-1,bgap1_3-1,bgap1_4-1,bgap1_5-
1,egap1_1+1,egap1_2+1,egap1_3+1,egap1_4+1,egap1_5+1)
    to (egap1_6);
    job1wks=-3;
    gpfl1_6=1;
  end;
end;
*** Remove gap 6 on job 1 - end gap date bad;
if bgap1_6>0 & egfl1_6=1 then do;
  do i=(bgap1_6) to (stopw1);
    job1wks=-3;
    gpfl1_6=1;
  end;
end;
*** Remove gap 6 on job 1 - both gap dates bad;
if bgfl1_6=1 & egfl1_6=1 then do;
  do i=max(starw1,bgap1_1-1,bgap1_2-1,bgap1_3-1,bgap1_4-1,bgap1_5-
1,egap1_1+1,egap1_2+1,egap1_3+1,egap1_4+1,egap1_5+1)
    to min(stopw1,bgap1_6-1,egap1_6+1);
    job1wks=-3;
    gpfl1_6=1;
  end;
end;
end; /* [1] */

if starfl_1=1 then do;
  do i=(starw1) to min(stopw1,bgap1_1-1,bgap1_2-1,bgap1_3-1,bgap1_4-1,bgap1_5-1,bgap1_6-1,
    egap1_1+1,egap1_2+1,egap1_3+1,egap1_4+1,egap1_5+1,egap1_6+1);
    job1wks=-3;
  end;
end;
if stopfl_1=1 then do;
  do i=max(stopw1,bgap1_1-1,bgap1_2-1,bgap1_3-1,bgap1_4-1,bgap1_5-1,bgap1_6-1,
    egap1_1+1,egap1_2+1,egap1_3+1,egap1_4+1,egap1_5+1,egap1_6+1)
    to (stopw1);
    job1wks=-3;
  end;
end;
endsas;

```

/\*At this point the program repeats the above code for jobs 2-9. This code is omitted due to space considerations. For more information, please contact NLS User Services.\*/

---

## Number of Weeks Worked during 19xx/20xx

**Variables Created:** CV\_WKSWK\_YR\_ALL.80 - CV\_WKSWK\_YR\_ALL.02  
CV\_WKSWK\_YR\_ET.80 - CV\_WKSWK\_YR\_ET.02  
CV\_WKSWK\_YR\_SE.98 - CV\_WKSWK\_YR\_SE.02

### Programs Used

This program uses [emp\\_begin.sas](#) as input (see the first page of this appendix for details).

This program counts the number of weeks each respondent worked at any job for each year of potential work activity (1980-2002). Three sets of variables are created: one for all jobs, one for employee-type jobs, and one for self-employed jobs. **See the [introduction to this appendix for important information about the universes for each of these variables](#).** Respondents not working in a given year are given a default value of zero (0) weeks worked. Otherwise, the variable indicates the actual cumulative weeks worked on all jobs in that year.

---

/\* Section 1: Create variables for # of weeks worked at any job during a given year. \*/

```
array job1wks (i) wk1_1-wk1_1200;
array job2wks (i) wk2_1-wk2_1200;
array job3wks (i) wk3_1-wk3_1200;
array job4wks (i) wk4_1-wk4_1200;
array job5wks (i) wk5_1-wk5_1200;
array job6wks (i) wk6_1-wk6_1200;
array job7wks (i) wk7_1-wk7_1200;
array job8wks (i) wk8_1-wk8_1200;
array alljobs (i) wks1-wks1200;
array selfjobs (i) swks1-swks1200;
array empjobs (i) ewks1-ewks1200;
array starw (i) starw1-starw8;
array stopw (i) stopw1-stopw8;
array starfl (i) starfl_1-starfl_8;
array stopfl (i) stopfl_1-stopfl_8;
array uid (i) uid1-uid8;
array self (i) self1-self8;
array r5wks (i) r5wks80-r5wks99 r5wks00 r5wks01 r5wks02;
array r5swks (i) r5swks80-r5swks99 r5swks00 r5swks01 r5swks02;
array r5ewks (i) r5ewks80-r5ewks99 r5ewks00 r5ewks01 r5ewks02;

do over alljobs;
  alljobs=0;
  selfjobs=0;
  empjobs=0;
end;

** Overlay multiple jobs over JOB 1 work weeks    **;
do over job1wks;
```

```

alljobs=job1wks;
if self1=1 then do; selfjobs=job1wks; end;
  if self1=0 then do; empjobs=job1wks; end;
end;

do over alljobs;
  if job2wks=1 then do;
    alljobs=job2wks;
  end;
  if job2wks=1 and self2=1 then do;
    selfjobs=job2wks;
  end;
  if job2wks=1 and self2=0 then do;
    empjobs=job2wks;
  end;

  if job2wks=-3 and alljobs=0 then do;
    alljobs=job2wks;
  end;
  if job2wks=-3 and selfjobs=0 and self2=1 then do;
    selfjobs=job2wks;
  end;
  if job2wks=-3 and empjobs=0 and self2=0 then do;
    empjobs=job2wks;
  end;
end;

```

/\* At this point the program loops through the job2 code for jobs 3-8. This code is omitted here.  
Contact NLS User Services for more information.\*/

\*\* Calculate cumulative weeks on all jobs for each year \*\*;

/\* Initialize each annual c.v. \*/  
do over r5wks; r5wks=0; r5swks=0; r5ewks=0; end;

/\* 1980 \*/

```

do i=1 to 52;
  if alljobs=1 then do; r5wks80=r5wks80+1; end;
  if selfjobs=1 then do; r5swks80=r5swks80+1; end;
  if empjobs=1 then do; r5ewks80=r5ewks80+1; end;
end;
do i=1 to 52;
  if alljobs=-3 then do; r5wks80=-3; end;
  if selfjobs=-3 then do; r5swks80=-3; end;
  if empjobs=-3 then do; r5ewks80=-3; end;
end;

```

\*\*\*\*\*At this point the program loops through the same code used above for 1980 for each year 1981-2002, creating the variables r4wks81, r4swks81, and r4ewks81; r4wks81, r4swks81, and r4ewks81; and so on through r4wks02, r4swks02, and r4ewks02. These loops are deleted due to space

considerations; users who need the entire code should contact NLS User Services. The week numbers for the "do i" statement for each year are as follows:

1981	53-104	1992	627-678
1982	105-156	1993	679-730
1983	157-209	1994	731-783
1984	210-261	1995	784-835
1985	262-313	1996	836-887
1986	314-365	1997	888-939
1987	366-417	1998	940-991
1988	418-470	1999	992-1044
1989	471-522	2000	1045-1096
1990	523-574	2001	1097-1148
1991	575-626	2002	1149-1200***/

```

do i=1 to 8;
/* start date invalid */
if starfl=1 then do;
  if stopw>1 and self=1 and r5wks80>0 then do; r5wks80=-3; r5swks80=-3; end;
  if stopw>1 and self=0 and r5wks80>0 then do; r5wks80=-3; r5ewks80=-3; end;
  if stopw>52 and self=1 and r5wks81>0 then do; r5wks81=-3; r5swks81=-3; end;
  if stopw>52 and self=0 and r5wks81>0 then do; r5wks81=-3; r5ewks81=-3; end;
  if stopw>104 and self=1 and r5wks82>0 then do; r5wks82=-3; r5swks82=-3; end;
  if stopw>104 and self=0 and r5wks82>0 then do; r5wks82=-3; r5ewks82=-3; end;
  if stopw>156 and self=1 and r5wks83>0 then do; r5wks83=-3; r5swks83=-3; end;
  if stopw>156 and self=0 and r5wks83>0 then do; r5wks83=-3; r5ewks83=-3; end;
  if stopw>209 and self=1 and r5wks84>0 then do; r5wks84=-3; r5swks84=-3; end;
  if stopw>209 and self=0 and r5wks84>0 then do; r5wks84=-3; r5ewks84=-3; end;
  if stopw>261 and self=1 and r5wks85>0 then do; r5wks85=-3; r5swks85=-3; end;
  if stopw>261 and self=0 and r5wks85>0 then do; r5wks85=-3; r5ewks85=-3; end;
  if stopw>313 and self=1 and r5wks86>0 then do; r5wks86=-3; r5swks86=-3; end;
  if stopw>313 and self=0 and r5wks86>0 then do; r5wks86=-3; r5ewks86=-3; end;
  if stopw>365 and self=1 and r5wks87>0 then do; r5wks87=-3; r5swks87=-3; end;
  if stopw>365 and self=0 and r5wks87>0 then do; r5wks87=-3; r5ewks87=-3; end;
  if stopw>417 and self=1 and r5wks88>0 then do; r5wks88=-3; r5swks88=-3; end;
  if stopw>417 and self=0 and r5wks88>0 then do; r5wks88=-3; r5ewks88=-3; end;
  if stopw>470 and self=1 and r5wks89>0 then do; r5wks89=-3; r5swks89=-3; end;
  if stopw>470 and self=0 and r5wks89>0 then do; r5wks89=-3; r5ewks89=-3; end;
  if stopw>522 and self=1 and r5wks90>0 then do; r5wks90=-3; r5swks90=-3; end;
  if stopw>522 and self=0 and r5wks90>0 then do; r5wks90=-3; r5ewks90=-3; end;
  if stopw>574 and self=1 and r5wks91>0 then do; r5wks91=-3; r5swks91=-3; end;
  if stopw>574 and self=0 and r5wks91>0 then do; r5wks91=-3; r5ewks91=-3; end;
  if stopw>626 and self=1 and r5wks92>0 then do; r5wks92=-3; r5swks92=-3; end;
  if stopw>626 and self=0 and r5wks92>0 then do; r5wks92=-3; r5ewks92=-3; end;
  if stopw>678 and self=1 and r5wks93>0 then do; r5wks93=-3; r5swks93=-3; end;
  if stopw>678 and self=0 and r5wks93>0 then do; r5wks93=-3; r5ewks93=-3; end;
  if stopw>730 and self=1 and r5wks94>0 then do; r5wks94=-3; r5swks94=-3; end;
  if stopw>730 and self=0 and r5wks94>0 then do; r5wks94=-3; r5ewks94=-3; end;

```

```

if stopw>783 and self=1 and r5wks95>0 then do; r5wks95=-3; r5swks95=-3; end;
if stopw>783 and self=0 and r5wks95>0 then do; r5wks95=-3; r5ewks95=-3; end;
if stopw>834 and self=1 and r5wks96>0 then do; r5wks96=-3; r5swks96=-3; end;
if stopw>834 and self=0 and r5wks96>0 then do; r5wks96=-3; r5ewks96=-3; end;
if stopw>887 and self=1 and r5wks97>0 then do; r5wks97=-3; r5swks97=-3; end;
if stopw>887 and self=0 and r5wks97>0 then do; r5wks97=-3; r5ewks97=-3; end;
if stopw>939 and self=1 and r5wks98>0 then do; r5wks98=-3; r5swks98=-3; end;
if stopw>939 and self=0 and r5wks98>0 then do; r5wks98=-3; r5ewks98=-3; end;
if stopw>990 and self=1 and r5wks99>0 then do; r5wks99=-3; r5swks99=-3; end;
if stopw>990 and self=0 and r5wks99>0 then do; r5wks99=-3; r5ewks99=-3; end;
if stopw>1043 and self=1 and r5wks00>0 then do; r5wks00=-3; r5swks00=-3; end;
if stopw>1043 and self=0 and r5wks00>0 then do; r5wks00=-3; r5ewks00=-3; end;
if stopw>1096 and self=1 and r5wks01>0 then do; r5wks01=-3; r5swks01=-3; end;
if stopw>1096 and self=0 and r5wks01>0 then do; r5wks01=-3; r5ewks01=-3; end;
if stopw>1148 and self=1 and r5wks02>0 then do; r5wks02=-3; r5swks02=-3; end;
if stopw>1148 and self=0 and r5wks02>0 then do; r5wks02=-3; r5ewks02=-3; end;
end;

```

/\* stop date invalid \*/

```

if stopfl=1 then do;
  if starw<53 and self=1 and r5wks80>0 then do; r5wks80=-3; r5swks80=-3; end;
  if starw<53 and self=0 and r5wks80>0 then do; r5wks80=-3; r5ewks80=-3;
end;
  if starw<105 and self=1 and r5wks81>0 then do; r5wks81=-3; r5swks81=-3; end;
  if starw<105 and self=0 and r5wks81>0 then do; r5wks81=-3; r5ewks81=-3;
end;
  if starw<157 and self=1 and r5wks82>0 then do; r5wks82=-3; r5swks82=-3; end;
  if starw<157 and self=0 and r5wks82>0 then do; r5wks82=-3; r5ewks82=-3;
end;
  if starw<210 and self=1 and r5wks83>0 then do; r5wks83=-3; r5swks83=-3; end;
  if starw<210 and self=0 and r5wks83>0 then do; r5wks83=-3; r5ewks83=-3;
end;
  if starw<262 and self=1 and r5wks84>0 then do; r5wks84=-3; r5swks84=-3; end;
  if starw<262 and self=0 and r5wks84>0 then do; r5wks84=-3; r5ewks84=-3; end;

  if starw<314 and self=1 and r5wks85>0 then do; r5wks85=-3; r5swks85=-3; end;
  if starw<314 and self=0 and r5wks85>0 then do; r5wks85=-3; r5ewks85=-3; end;
  if starw<366 and self=1 and r5wks86>0 then do; r5wks86=-3; r5swks86=-3; end;
  if starw<366 and self=0 and r5wks86>0 then do; r5wks86=-3; r5ewks86=-3; end;
  if starw<418 and self=1 and r5wks87>0 then do; r5wks87=-3; r5swks87=-3; end;
  if starw<418 and self=0 and r5wks87>0 then do; r5wks87=-3; r5ewks87=-3; end;
  if starw<471 and self=1 and r5wks88>0 then do; r5wks88=-3; r5swks88=-3; end;
  if starw<471 and self=0 and r5wks88>0 then do; r5wks88=-3; r5ewks88=-3; end;
  if starw<523 and self=1 and r5wks89>0 then do; r5wks89=-3; r5swks89=-3; end;
  if starw<523 and self=0 and r5wks89>0 then do; r5wks89=-3; r5ewks89=-3; end;
  if starw<575 and self=1 and r5wks90>0 then do; r5wks90=-3; r5swks90=-3; end;
  if starw<575 and self=0 and r5wks90>0 then do; r5wks90=-3; r5ewks90=-3; end;
  if starw<627 and self=1 and r5wks91>0 then do; r5wks91=-3; r5swks91=-3; end;
  if starw<627 and self=0 and r5wks91>0 then do; r5wks91=-3; r5ewks91=-3; end;
  if starw<679 and self=1 and r5wks92>0 then do; r5wks92=-3; r5swks92=-3; end;
  if starw<679 and self=0 and r5wks92>0 then do; r5wks92=-3; r5ewks92=-3; end;

```

```

if starw<731 and self=1 and r5wks93>0 then do; r5wks93=-3; r5swks93=-3; end;
if starw<731 and self=0 and r5wks93>0 then do; r5wks93=-3; r5ewks93=-3; end;
if starw<784 and self=1 and r5wks94>0 then do; r5wks94=-3; r5swks94=-3; end;
if starw<784 and self=0 and r5wks94>0 then do; r5wks94=-3; r5ewks94=-3; end;
if starw<836 and self=1 and r5wks95>0 then do; r5wks95=-3; r5swks95=-3; end;
if starw<836 and self=0 and r5wks95>0 then do; r5wks95=-3; r5ewks95=-3; end;
if starw<888 and self=1 and r5wks96>0 then do; r5wks96=-3; r5swks96=-3; end;
if starw<888 and self=0 and r5wks96>0 then do; r5wks96=-3; r5ewks96=-3; end;
if starw<940 and self=1 and r5wks97>0 then do; r5wks97=-3; r5swks97=-3; end;
if starw<940 and self=0 and r5wks97>0 then do; r5wks97=-3; r5ewks97=-3; end;
if starw<991 and self=1 and r5wks98>0 then do; r5wks98=-3; r5swks98=-3; end;
if starw<991 and self=0 and r5wks98>0 then do; r5wks98=-3; r5ewks98=-3; end;
if starw<1044 and self=1 and r5wks99>0 then do; r5wks99=-3; r5swks99=-3; end;
if starw<1044 and self=0 and r5wks99>0 then do; r5wks99=-3; r5ewks99=-3; end;
if starw<1096 and self=1 and r5wks00>0 then do; r5wks00=-3; r5swks00=-3; end;
if starw<1096 and self=0 and r5wks00>0 then do; r5wks00=-3; r5ewks00=-3; end;
if starw<1149 and self=1 and r5wks01>0 then do; r5wks01=-3; r5swks01=-3; end;
if starw<1149 and self=0 and r5wks01>0 then do; r5wks01=-3; r5ewks01=-3; end;
if starw<1201 and self=1 and r5wks02>0 then do; r5wks02=-3; r5swks02=-3; end;
if starw<1201 and self=0 and r5wks02>0 then do; r5wks02=-3; r5ewks02=-3; end;
end;

```

```

*** Include valid skips;
do over r5wks;
  if e200=-5 then do;
    r5wks=-5;
    r5swks=-5;
    r5ewks=-5;
  end;
end;

```

```

/* Define the following c.v.'s to match up the array sizes, they are zero by definition. */
r1wks99=0; r1wks00=0; r1wks01=0; r1wks02=0;
r2wks00=0; r2wks01=0; r2wks02=0;
r3wks01=0; r3wks02=0;

```

```

array r1wks (i) r1wks80-r1wks99 r1wks00 r1wks01 r1wks02; /* Round 1 created variable */
array r2wks (i) r2wks80-r2wks99 r2wks00 r2wks01 r2wks02; /* Round 2 created variable */
array r3wks (i) r3wks80-r3wks99 r3wks00 r3wks01 r3wks02; /* Round 3 created variable */
array r4wks (i) r4wks80-r4wks99 r4wks00 r4wks01 r4wks02; /* Round 4 created variable, all jobs */
array r4swks (i) r4swks80-r4swks99 r4swks00 r4swks01 r4wks02; /* Round 4 created variable, self-employed jobs */
array r4ewks (i) r4ewks80-r4ewks99 r4ewks00 r4ewks01 r4wks02; /* Round 4 created variable, employee jobs */
array r5wks (i) r5wks80-r5wks99 r5wks00 r5wks01 r5wks02; /* Round 5 created variable, all jobs */
array r5swks (i) r5swks80-r5swks99 r5swks00 r5swks01 r5wks02; /* Round 5 created variable, self-employed jobs */
array r5ewks (i) r5ewks80-r5ewks99 r5ewks00 r5ewks01 r5wks02; /* Round 5 created variable, employee jobs */
array twks (i) twks80-twks99 twks00 twks01 twks02; /* Total created variable, all jobs */
array tswks (i) tswks80-tswks99 tswks00 tswks01 tswks02; /* Total created variable, self-employed

```

```

jobs */
array tewks (i) tewks80-tewks99 tewks00 tewks01 tewks02; /* Total created variable, employee jobs */

/* Thes people backreported on a military job. The problem is that the backreporting skips them into the
bottom calculation of the c.v.'s, so their "back4" is set back to ". ". The exception is pubid 5977, who
backreported a military job and a employee job. There are 8 respondents. */

array br (i) br1-br9;
array milflag (i) milflag1-milflag9;
do over br;
  if br=1 and milflag=1 and pubid ne 5977 then do;
    back4=.;
  end;
end;

/* Begin by splitting up periods where Round4 the dli c.v. exclusively collect weeks worked
information. Define the dli interview year as the split. By construction, any weeks worked information
collected before the dli interview week should be independent of information collected in Round4. */
/* For the new Round 4 c.v.'s, we are assuming that all dli jobs are employee-type jobs. Thus, dli
information will be added to the "all jobs" and "employee jobs" c.v.'s but not "self-employed jobs" c.v.'s.
*/
/* dli is Round 4 */

do over r5wks;
  twks=0;
  tswks=0;
  tewks=0;
  if r5wks>0 then do; twks=r5wks; end;
  if r5swks>0 then do; tswks=r5swks; end;
  if r5ewks>0 then do; tewks=r5ewks; end;
  if r4wks>0 and dliwk=r4int then do;
    twks=r4wks;
  end;
  if r4swks>0 and dliwk=r4int then do;
    tswks=r4swks;
  end;
  if r4ewks>0 and dliwk=r4int then do;
    tewks=r4ewks;
  end;
  if r4wks>0 and r5wks>0 and dliwk=r4int then do;
    twks=r4wks+r5wks;
  end;
  if r4swks>0 and r5swks>0 and dliwk=r4int then do;
    tswks=r4swks+r5swks;
  end;
  if r4ewks>0 and r5ewks>0 and dliwk=r4int then do;
    tewks=r4ewks+r5ewks;
  end;
  /* dli is Round 3 */
  if r3wks>0 and dliwk=r3int then do;
    twks=r3wks;

```

```

    tewks=r3wks;
end;
if r3wks>0 and r5wks>0 and dliwk=r3int then do;
    twks=r3wks+r5wks;
end;
if r3wks>0 and r5ewks>0 and dliwk=r3int then do;
    tewks=r3wks+r5ewks;
end;
/* dli is Round 2 */
if r2wks>0 and dliwk=r2int then do;
    twks=r2wks;
    tewks=r2wks;
end;
if r2wks>0 and r5wks>0 and dliwk=r2int then do;
    twks=r2wks+r5wks;
end;
if r2wks>0 and r5ewks>0 and dliwk=r2int then do;
    tewks=r2wks+r5ewks;
end;
/* dli is Round 1 */
if r1wks>0 and dliwk=r1int then do;
    twks=r1wks;
    tewks=r1wks;
end;
if r1wks>0 and r5wks>0 and dliwk=r1int then do;
    twks=r1wks+r5wks;
end;
if r1wks>0 and r5ewks>0 and dliwk=r1int then do;
    tewks=r1wks+r5ewks;
end;
end;

```

/\* This section is written for the backreporters. We load in the weeks array from the Round 4 c.v.'s and use it to update the corresponding Round 5 array. \*/

```

array alljobs (i) wks1-wks1200;
array selfjobs (i) swks1-swks1200;
array empjobs (i) ewks1-ewks1200;
array r4arr (i) r4ar1-r4ar1200;

```

```

if back4=1 then do;
do over alljobs;
    if r4arr=1 then do;
        alljobs=r4arr;
    end;
    if r4arr=1 and self2=0 then do;
        empjobs=r4arr;
    end;

```

```

if r4arr=-3 and alljobs=0 then do;
    alljobs=r4arr;

```

```

end;
if r4arr=-3 and empjobs=0 and self2=0 then do;
  empjobs=r4arr;
end;
end;

/* 1990 */
twks90=0;
tswks90=0;
tewks90=0;

do i=523 to 574;
  if alljobs=1 then do; twks90=twks90+1; end;
  if selfjobs=1 then do; tswks90=tswks90+1; end;
  if empjobs=1 then do; tewks90=tewks90+1; end;
end;
do i=523 to 574;
  if alljobs=-3 then do; twks90=-3; end;
  if selfjobs=-3 then do; tswks90=-3; end;
  if empjobs=-3 then do; tewks90=-3; end;
end;

```

\*\*\*\*\*At this point the program loops through the same code used above for 1990 for each year 1991-2001. These loops are deleted due to space considerations; users who need the entire code should contact NLS User Services. The week numbers for the "do i" statement for each year are as follows:

1991	575-626
1992	627-678
1993	679-730
1994	731-783
1995	784-835
1996	836-887
1997	888-939
1998	940-991
1999	992-1044
2000	1045-1096
2001	1097-1148***

```

/* 2002 */
twks02=0;
tswks02=0;
tewks02=0;
do i=1149 to 1200;
  if alljobs=1 then do; twks02=twks02+1; end;
  if selfjobs=1 then do; tswks02=tswks02+1; end;
  if empjobs=1 then do; tewks02=tewks02+1; end;
end;
do i=1149 to 1200;
  if alljobs=-3 then do; twks02=-3; end;

```

```

if selfjobs=-3 then do; tswks02=-3; end;
  if empjobs=-3 then do; tewks02=-3; end;
end;
end;

/* If either created variable from Round 5 or dli is a -2 or -3, then the overall created variable will be -2
or -3,
respectively. */
do over r4wks;
  if r5wks in (-1,-2,-3) then do;
    twks=-3;
  end;
  if r5swks in (-1,-2,-3) then do;
    tswks=-3;
  end;
  if r5ewks in (-1,-2,-3) then do;
    tewks=-3;
  end;
  if r4wks in (-1,-2,-3) and dliwk=r4int then do;
    twks=-3;
  end;
  if r4swks in (-1,-2,-3) and dliwk=r4int then do;
    tswks=-3;
  end;
  if r4ewks in (-1,-2,-3) and dliwk=r4int then do;
    tewks=-3;
  end;
  if r3wks in (-1,-2,-3) and dliwk=r3int then do;
    twks=-3;
    tewks=-3;
  end;
  if r2wks in (-1,-2,-3) and dliwk=r2int then do;
    twks=-3;
    tewks=-3;
  end;
  if r1wks in (-1,-2,-3) and dliwk=r1int then do;
    twks=-3;
    tewks=-3;
  end;
end;

/* Round 5 non-interview cases */
do over twks;
  if intwk=-5 then do;
    twks=-5;
    tswks=-5;
    tewks=-5;
  end;
end;

/* Backreporter hand edits. The only backreporters we need to worry about are the ones that missed

```

Round 4 interview. Backreporters who were interviewed in Round 4 are accounted for in the "r4arr" array section. Because of the small number of missing interview backreporters (15 total, with 8 needing a hand edit), it was easier to hand edit them instead of loading in the dli arrays. \*/

```
if pubid=2685 then do; twks99=52; tewks99=52; end;
if pubid=1429 then do; twks00=53; tewks00=53; end;
if pubid=1910 then do; twks00=24; tewks00=24; end;
if pubid=2478 then do; twks00=-3; tewks00=-3; end;
if pubid=5466 then do; twks00=22; tewks00=22; end;
if pubid=3460 then do; twks00=30; tewks00=0; end;
if pubid=7224 then do; twks00=28; tewks00=16; end;

endsas;
```

---

## Number of Weeks Worked since Last Interview

**Variables Created:** CV\_WKSWK\_DLI\_ALL  
CV\_WKSWK\_DLI\_ET  
CV\_WKSWK\_DLI\_SE

### Programs Used

This program uses [emp\\_begin.sas](#) as input (see the first page of this appendix for details).

For each individual, this program counts the number of weeks the respondent worked since the last interview. Three variables are created: one for all jobs, one for employee-type jobs, and one for self-employed jobs. **See the [introduction to this appendix for important information about the universes for each of these variables](#).** Respondents not working in a given year are given a default value of zero.

---

```
array job1wks (i) wk1_1-wk1_1200;
array job2wks (i) wk2_1-wk2_1200;
array job3wks (i) wk3_1-wk3_1200;
array job4wks (i) wk4_1-wk4_1200;
array job5wks (i) wk5_1-wk5_1200;
array job6wks (i) wk6_1-wk6_1200;
array job7wks (i) wk7_1-wk7_1200;
array job8wks (i) wk8_1-wk8_1200;
array alljobs (i) wks1-wks1200;
array selfjobs (i) swks1-swks1200;
array empjobs (i) ewks1-ewks1200;
array starw (i) starw1-starw8;
array stopw (i) stopw1-stopw8;
array self (i) self1-self8;

do over alljobs; alljobs=0; selfjobs=0; empjobs=0; end;
```

```
** Overlay multiple jobs over JOB 1 work weeks      **;
```

```
do over job1wks;  
  alljobs=job1wks;  
  if self1=1 then do; selfjobs=job1wks; end;  
  if self1=0 then do; empjobs=job1wks; end;  
end;
```

```
do over alljobs;  
  if job2wks=1 then do;  
    alljobs=job2wks;  
  end;  
  if job2wks=1 and self2=1 then do;  
    selfjobs=job2wks;  
  end;  
  if job2wks=1 and self2=0 then do;  
    empjobs=job2wks;  
  end;
```

```
if job2wks=-3 and alljobs=0 then do;  
  alljobs=job2wks;  
end;  
if job2wks=-3 and selfjobs=0 and self2=1 then do;  
  selfjobs=job2wks;  
end;  
if job2wks=-3 and empjobs=0 and self2=0 then do;  
  empjobs=job2wks;  
end;  
end;
```

```
/*At this point the program loops through the job2 code for jobs 3-8. Contact NLS User Services for details.*/
```

```
** Calculate cumulative weeks on all jobs since dli **;  
allwks=0;  
selfwks=0;  
empwks=0;
```

```
if dliwk>0 then do;  
  if dliwk=>age14wk then do;  
    do i=dliwk to 1200;  
      if alljobs=1 then do; allwks=allwks+1; end;  
      if selfjobs=1 then do; selfwks=selfwks+1; end;  
      if empjobs=1 then do; empwks=empwks+1; end;  
    end;  
    do i=dliwk to 1200;  
      if alljobs=-3 then do; allwks=-3; end;  
      if selfjobs=-3 then do; selfwks=-3; end;  
      if empjobs=-3 then do; empwks=-3; end;  
    end;  
  end;
```

```

if age14wk>dliwk then do;
do i=age14wk to 1200;
  if alljobs=1 then do; allwks=allwks+1; end;
  if selfjobs=1 then do; selfwks=selfwks+1; end;
  if empjobs=1 then do; empwks=empwks+1; end;
end;
do i=age14wk to 1200;
  if alljobs=-3 then do; allwks=-3; end;
  if selfjobs=-3 then do; selfwks=-3; end;
  if empjobs=-3 then do; empwks=-3; end;
end;
end;
end;

if starfl_1=1 then do; allwks=-3; end;
if starfl_2=1 then do; allwks=-3; end;
if starfl_3=1 then do; allwks=-3; end;
if starfl_4=1 then do; allwks=-3; end;
if starfl_5=1 then do; allwks=-3; end;
if starfl_6=1 then do; allwks=-3; end;
if starfl_7=1 then do; allwks=-3; end;
if starfl_8=1 then do; allwks=-3; end;

if stopfl_1=1 then do; allwks=-3; end;
if stopfl_2=1 then do; allwks=-3; end;
if stopfl_3=1 then do; allwks=-3; end;
if stopfl_4=1 then do; allwks=-3; end;
if stopfl_5=1 then do; allwks=-3; end;
if stopfl_6=1 then do; allwks=-3; end;
if stopfl_7=1 then do; allwks=-3; end;
if stopfl_8=1 then do; allwks=-3; end;

if e200=-5 then do;
  allwks=-5;
  selfwks=-5;
  empwks=-5;
end;
endsas;

```

---

## **Number of Weeks Worked Age 14-19, Weeks Worked at All Jobs Since Age 20, and Weeks Worked at Employee-Type Jobs Since Age 20**

**Variables Created:** CV\_WKSWK\_TEEN  
 CV\_WKSWK\_ADULT\_ET  
 CV\_WKSWK\_ADULT\_ALL

## **Programs Used**

This program uses [\*\*emp\\_begin.sas\*\*](#) as input (see the first page of this appendix for details).

For each respondent, this program creates a variable counting the number of weeks worked at any employee-type job between the ages of 14 and 19. It then creates two variables counting weeks worked from age 20 to the present; one totals all jobs and one presents only employee jobs. **See the introduction to this appendix for important information about the universes for each of these variables.** Respondents not working are given a default value of zero; respondents who have not yet reached age 20 receive a valid skip (-4) on the latter two variables.

---

```
array job1wks (i) wk1_1-wk1_1200;
array job2wks (i) wk2_1-wk2_1200;
array job3wks (i) wk3_1-wk3_1200;
array job4wks (i) wk4_1-wk4_1200;
array job5wks (i) wk5_1-wk5_1200;
array job6wks (i) wk6_1-wk6_1200;
array job7wks (i) wk7_1-wk7_1200;
array job8wks (i) wk8_1-wk8_1200;
array job9wks (i) wk9_1-wk9_1200;
array alljobs (i) wks1-wks1200;
array empjobs (i) ewks1-ewks1200;
array starw (i) starw1-starw8;
array stopw (i) stopw1-stopw8;
array starfl (i) starfl_1-starfl_8;
array stopfl (i) stopfl_1-stopfl_8;

** Overlay multiple jobs over JOB 1 work weeks      **;

do over empjobs; empjobs=0; end;

do over job1wks;
  if self1=0 then do; empjobs=job1wks; end;
end;

do over empjobs;
  if job2wks=1 and self2=0 then do;
    empjobs=job2wks;
  end;
  if job2wks=-3 and self2=0 and empjobs=0 then do;
    empjobs=job2wks;
  end;
end;

/*Repeat the job 2 code above for jobs 3-8; contact NLS User Services for more information.*/

do over alljobs; alljobs=0; end;

do over job1wks; alljobs=job1wks; end;

do over alljobs;
  if job2wks=1 then do;
```

```

    alljobs=job2wks;
end;
if job2wks=-3 and alljobs=0 then do;
    alljobs=job2wks;
end;
end;

/*Repeat the job 2 code above for jobs 3-8; contact NLS User Services for more information.*/

** Calculate cumulative weeks on all jobs since age 14 ** ;

/* r5emp14 is the variable that counts the number of weeks worked since respondent's 14th birthday. */
r5emp14=0;
r5wks20=0;
r5emp20=0;

/* Note the max for age20wk is 1305. */
if age14wk>0 and (age20wk-1)<=1200 then do;
    do i=age14wk to (age20wk-1);
        if empjobs=1 then do; r5emp14=r5emp14+1; end;
    end;
    do i=age14wk to (age20wk-1);
        if empjobs=-3 then do; r5emp14=-3; end;
    end;
end;

if age14wk>0 and (age20wk-1)>1200 then do;
    do i=age14wk to 1200;
        if empjobs=1 then do; r5emp14=r5emp14+1; end;
    end;
    do i=age14wk to 1200;
        if empjobs=-3 then do; r5emp14=-3; end;
    end;
end;

if age14wk>0 and age20wk<=1200 then do;
    do i=age20wk to 1200;
        if alljobs=1 then do; r5wks20=r5wks20+1; end;
    end;
    do i=age20wk to 1200;
        if alljobs=-3 then do; r5wks20=-3; end;
    end;
end;

if age14wk>0 and age20wk<=1200 then do;
    do i=age20wk to 1200;
        if empjobs=1 then do; r5emp20=r5emp20+1; end;
    end;
    do i=age20wk to 1200;
        if empjobs=-3 then do; r5emp20=-3; end;
    end;
end;

```

```

end;

/*the lowest age14wk is 731, the lowest age20wk is 1044*/
do i=1 to 8;
if age14wk<784 then do;
  if starfl=1 and r5emp14>0 and stopw>730 then do;
    r5emp14=-3;
  end;
end;
if age14wk<836 then do;
  if starfl=1 and r5emp14>0 and stopw>783 then do;
    r5emp14=-3;
  end;
end;
if age14wk<888 then do;
  if starfl=1 and r5emp14>0 and stopw>835 then do;
    r5emp14=-3;
  end;
end;
if age14wk<940 then do;
  if starfl=1 and r5emp14>0 and stopw>887 then do;
    r5emp14=-3;
  end;
end;
if age14wk<991 then do;
  if starfl=1 and r5emp14>0 and stopw>939 then do;
    r5emp14=-3;
  end;
end;
if age14wk<1044 then do;
  if starfl=1 and r5emp14>0 and stopw>990 then do;
    r5emp14=-3;
  end;
end;
if age14wk<1097 then do;
  if starfl=1 and r5emp14>0 and stopw>1043 then do;
    r5emp14=-3;
  end;
end;
if age20wk<1097 then do;
  if starfl=1 and r5wks20>0 and stopw>1043 then do;
    r5wks20=-3;
    r5emp20=-3;
  end;
end;
if age14wk<1149 then do;
  if starfl=1 and r5emp14>0 and stopw>1096 then do;
    r5emp14=-3;
  end;
end;
if age20wk<1149 then do;

```

```

if starfl=1 and r5wks20>0 and stopw>1096 then do;
  r5wks20=-3;
  r5emp20=-3;
end;
end;
if age14wk<1201 then do;
  if starfl=1 and r5emp14>0 and stopw>1148 then do;
    r5emp14=-3;
  end;
end;
if age20wk<1201 then do;
  if starfl=1 and r5wks20>0 and stopw>1148 then do;
    r5wks20=-3;
    r5emp20=-3;
  end;
end;
if age14wk<784 then do;
  if stopfl=1 and r5emp14>0 and starw<784 then do;
    r5emp14=-3;
  end;
end;
if age14wk<836 then do;
  if stopfl=1 and r5emp14>0 and starw<836 then do;
    r5emp14=-3;
  end;
end;
if age14wk<888 then do;
  if stopfl=1 and r5emp14>0 and starw<888 then do;
    r5emp14=-3;
  end;
end;
if age14wk<940 then do;
  if stopfl=1 and r5emp14>0 and starw<940 then do;
    r5emp14=-3;
  end;
end;
if age14wk<991 then do;
  if stopfl=1 and r5emp14>0 and starw<991 then do;
    r5emp14=-3;
  end;
end;
if age14wk<1044 then do;
  if stopfl=1 and r5emp14>0 and starw<1044 then do;
    r5emp14=-3;
  end;
end;
if age14wk<1097 then do;
  if stopfl=1 and r5emp14>0 and starw<1097 then do;
    r5emp14=-3;
  end;

```

```

end;
if age20wk<1097 then do;
  if stopfl=1 and r5wks20>0 and starw<1097 then do;
    r5wks20=-3;
    r5emp20=-3;
  end;
end;
if age14wk<1149 then do;
  if stopfl=1 and r5emp14>0 and starw<1149 then do;
    r5emp14=-3;
  end;
end;
if age20wk<1149 then do;
  if stopfl=1 and r5wks20>0 and starw<1149 then do;
    r5wks20=-3;
    r5emp20=-3;
  end;
end;
if age14wk<1201 then do;
  if stopfl=1 and r5emp14>0 and starw<1201 then do;
    r5emp14=-3;
  end;
end;
if age20wk<1201 then do;
  if stopfl=1 and r5wks20>0 and starw<1201 then do;
    r5wks20=-3;
    r5emp20=-3;
  end;
end;
end;

```

```

*** Include noninterviews;
if e200=-5 then do;
  r5emp14=-5;
  r5wks20=-5;
  r5emp20=-3;
end;

```

```

r4emp14=temp14;
r4wks20=twks20;
r4emp20=temp20;

```

```

/* Initialize created variable for both rounds to zero. */
temp14=0;
twks20=0;
temp20=0;

```

```

/* These people backreported on a military job. The problem is that the backreporting skips them into
the bottom calculation of the c.v.'s, so their "back4" is set back to ". ". The exception is pubid 5977, who
backreported a military job and a employee job. There are 8 of these people. */

```

```

array br (i) br1-br9;
array milflag (i) milflag1-milflag9;
do over br;
  if br=1 and milflag=1 and pubid ne 5977 then do;
    back4=.;
    end;
  end;

/* By the construction of the Round 5 created variable, we can add the two created variables from
Round 5 and dli Round if they are both positive. If one is positive and one is zero, then the positive value
will be the total created variable for both rounds. If neither is positive, then the total created variable for
both rounds will be zero. If either the Round 5 c.v. or the dli c.v. is invalid, then the total c.v. will be -3.
*/
if r5emp14=>0 then do; temp14=r5emp14; end;
if r5emp14=-3 then do; temp14=-3; end;

/* dli is Round 4 */
if r4emp14=>0 then do; temp14=r4emp14; end;
if r4emp14=>0 and r5emp14=>0 then do;
  temp14=r4emp14+r5emp14;
end;
if r4emp14=-3 then do; temp14=-3; end;

/* dli is Round 3 */
if r3wks14=>0 and r4int=-5 then do;
  temp14=r3wks14;
end;
if r3wks14=>0 and r5emp14=>0 and r4int=-5 then do;
  temp14=r3wks14+r5emp14;
end;
if r3wks14=-3 and r4int=-5 then do; temp14=-3; end;

/* dli is Round 2 */
if r2wks14=>0 and r4int=-5 and r3int=-5 then do;
  temp14=r2wks14;
end;
if r2wks14=>0 and r5emp14=>0 and r4int=-5 and r3int=-5 then do;
  temp14=r2wks14+r5emp14;
end;
if r2wks14=-3 and r4int=-5 and r3int=-5 then do;
  temp14=-3;
end;

/* dli is Round 1 */
if r1wks14=>0 and r4int=-5 and r3int=-5 and r2int=-5 then do;
  temp14=r1wks14;
end;
if r1wks14=>0 and r5emp14=>0 and r4int=-5 and r3int=-5 and r2int=-5 then do;
  temp14=r1wks14+r5emp14;

```

```

end;
if r1wks14=-3 and r4int=-5 and r3int=-5 and r2int=-5 then do; temp14=-3; end;

if r5wks20>0 then twks20=r5wks20;
if r4wks20>0 then twks20=r4wks20;
if r5wks20>0 and r4wks20>0 then do;
  twks20=r5wks20+r4wks20;
end;
if r5wks20=-3 or r4wks20=-3 then do;
  twks20=-3;
end;

if r5emp20>0 then temp20=r5emp20;
if r4emp20>0 then temp20=r4emp20;
if r5emp20>0 and r4emp20>0 then do;
  temp20=r5emp20+r4emp20;
end;
if r5emp20=-3 or r4emp20=-3 then do;
  temp20=-3;
end;

```

*/\* This section is written for the backreporters. We load in the weeks array from the Round 4 c.v.'s and use it to update the corresponding Round 5 jobs array. \*/*

```

array alljobs (i) wks1-wks1200;
array empjobs (i) ewks1-ewks1200;
array r4arr (i) r4ar1-r4ar1200;

```

*/\* There are some backreporters who are older than 20 at the Round 4 and 5 interviews. Their age14 variables should carry over to Round 5, but it will not since it is written over below by the Round 4 arrays which will not contain any age 14 to 19 job information. The following creates a dummy variable to identify these people (36 of them) which preserves their temp14 data compiled above. \*/*

```

array starw (i) starw1-starw8;
do over starw;
  if back4=1 and starw>age20wk and age20wk<r4int then do;
    oldback=1;
  end;
end;

```

```

if back4=1 then do;
  do over alljobs;
    if r4arr=1 then do;
      alljobs=r4arr;
      empjobs=r4arr;
    end;
    if r4arr=-3 and alljobs=0 then do;
      alljobs=r4arr;
    end;
    if r4arr=-3 and empjobs=0 then do;

```

```

empjobs=r4arr;
end;
end;

/* We want to keep the temp14 calculation for people older than 20 at Round 4, but the age20 c.v.'s need
to be reset for all Round 4 backreporters.*/
if oldback=. then do;
  temp14=0;
end;

twks20=0;
temp20=0;

/* People who are over 20 in Round 5. See above comment for info on "oldback".*/
if age14wk>0 and (age20wk-1)<=1200 and oldback=. then do;
  do i=age14wk to (age20wk-1);
    if empjobs=1 then do; temp14=temp14+1; end;
    end;
  do i=age14wk to (age20wk-1);
    if empjobs=-3 then do; temp14=-3; end;
    end;
  end;

if age14wk>0 and age20wk<=1200 then do;
  do i=age20wk to 1200;
    if alljobs=1 then do; twks20=twks20+1; end;
    end;
  do i=age20wk to 1200;
    if alljobs=-3 then do; twks20=-3; end;
    end;
  end;

if age14wk>0 and age20wk<=1200 then do;
  do i=age20wk to 1200;
    if empjobs=1 then do; temp20=temp20+1; end;
    end;
  do i=age20wk to 1200;
    if empjobs=-3 then do; temp20=-3; end;
    end;
  end;

/* People who are under 20 */
if age14wk>0 and (age20wk-1)>1200 then do;
  do i=age14wk to 1200;
    if empjobs=1 then do; temp14=temp14+1; end;
    end;
  do i=age14wk to 1200;
    if empjobs=-3 then do; temp14=-3; end;
    end;
  end;

```

end;

/\* Backreporters who were not interviewed in Round 2 will be hand-edits, since loading in their Round 3 weeks array is more work than looking 5 cases up. \*/

/\* These 2 backreporters are older than 20 at Round 4 and Round 5, but have start dates that predate age20wk and round 4 int. date. The above does not account for this type of data. \*/

if pubid=7014 then temp14=59;  
if pubid=3073 then temp14=-3;

if uid1=-5 then do;  
temp14=-5;  
twks20=-5;  
temp20=-5;  
end;

/\* For respondents younger than 20 at interview date, their thrs14 c.v. will be -4. \*/

if age20wk>intwk and intwk ne -5 then do;  
twks20=-4;  
temp20=-4;  
end;

/\* Correction of misread Round 4 data. \*/

if pubid in (176,3130,3527,3565,6312) then do;  
temp14=maxcv14;  
end;

if age20wk<intwk then do;  
maxcv20=intwk-age20wk+1;  
if twks20>maxcv20 and twks20=>0 then yuck=1;  
end;

/\* Here are the backreporter hand edits. The only backreporters we need to worry about are the ones that missed Round 4 interview. Backreporters who were interviewed in Round 4 are accounted for in the "r4arr" array section. Because of the small number of missing interview backreporters (15 total, with 6 needing a hand edit), it was easier to hand edit them instead of loading in the dli arrays. \*/

if pubid=1429 then temp14=229;  
if pubid=1910 then temp14=165;  
if pubid=2685 then temp14=198;  
if pubid=5466 then temp14=56;  
if pubid=7224 then temp14=101;  
if pubid=3460 then temp14=29;

endsas;

---

## Weeks Worked at Employee Job #x during 19xx/20xx

**Variables Created:** CV\_WKSWK\_JOB\_YR.01.80 - CV\_WKSWK\_JOB\_YR.01.02  
CV\_WKSWK\_JOB\_YR.02.80 - CV\_WKSWK\_JOB\_YR.02.02 etc. through job  
#9

## Programs Used

This program uses [emp\\_begin.sas](#) as input (see the first page of this appendix for details).

This program creates variables for each of the respondent's jobs counting the number of weeks worked in each calendar year. A variable is created for each potential job even if the respondent has worked no jobs in a given year with the default value set to zero (0). The most jobs held by any respondent in round 5 was eight, so variables are created for eight jobs for each respondent.

---

### **/\* Section 1: Create the Round 5 variable for number of weeks worked at a given job in a given year. \*/**

```
array starw (i) starw1-starw9;
array stopw (i) stopw1-stopw9;
array job1wks (i) wk1_1-wk1_1200;
array job2wks (i) wk2_1-wk2_1200;
array job3wks (i) wk3_1-wk3_1200;
array job4wks (i) wk4_1-wk4_1200;
array job5wks (i) wk5_1-wk5_1200;
array job6wks (i) wk6_1-wk6_1200;
array job7wks (i) wk7_1-wk7_1200;
array job8wks (i) wk8_1-wk8_1200;
array starfl (i) starfl_1-starfl_8;
array stopfl (i) stopfl_1-stopfl_8;

array r5wks80 (i) r5wks801-r5wks808;
array r5wks81 (i) r5wks811-r5wks818;
/* and so on through */
array r5wks99 (i) r5wks991-r5wks998;
array r5wks00 (i) r5wks001-r5wks008;
array r5wks01 (i) r5wks011-r5wks018;
array r5wks02 (i) r5wks021-r5wks028;

** Calculate cumulative weeks on individual jobs for each year **;
/* 1980 */

r5wks801=0; r5wks802=0; r5wks803=0; r5wks804=0; r5wks805=0; r5wks806=0; r5wks807=0;
r5wks808=0;

do i=1 to 52;
  if job1wks=1 then do; r5wks801=r5wks801+1; end;
  if job2wks=1 then do; r5wks802=r5wks802+1; end;
  if job3wks=1 then do; r5wks803=r5wks803+1; end;
  if job4wks=1 then do; r5wks804=r5wks804+1; end;
  if job5wks=1 then do; r5wks805=r5wks805+1; end;
  if job6wks=1 then do; r5wks806=r5wks806+1; end;
```

```

if job7wks=1 then do; r5wks807=r5wks807+1; end;
if job8wks=1 then do; r5wks808=r5wks808+1; end;

if job1wks=-3 then do; r5wks801=-3; end;
if job2wks=-3 then do; r5wks802=-3; end;
if job3wks=-3 then do; r5wks803=-3; end;
if job4wks=-3 then do; r5wks804=-3; end;
if job5wks=-3 then do; r5wks805=-3; end;
if job6wks=-3 then do; r5wks806=-3; end;
if job7wks=-3 then do; r5wks807=-3; end;
if job8wks=-3 then do; r5wks808=-3; end;
end;

```

/\*At this point the program repeats the above code for each year through 2001. Due to space considerations this code is not included here. Researchers needing the complete code should contact NLS User Services. The variables and "do i" statements for each year are as follows:

1981	wksw811-wksw818	do i=53 to 104
1982	wksw821-wksw828	do i=105 to 156
1983	wksw831-wksw838	do i=157 to 209
1984	wksw841-wksw848	do i=210 to 261
1985	wksw851-wksw858	do i=262 to 313
1986	wksw861-wksw868	do i=314 to 365
1987	wksw871-wksw878	do i=366 to 417
1988	wksw881-wksw888	do i=418 to 470
1989	wksw891-wksw898	do i=471 to 522
1990	wksw901-wksw908	do i=523 to 574
1991	wksw911-wksw918	do i=575 to 626
1992	wksw921-wksw928	do i=627 to 678
1993	wksw931-wksw938	do i=679 to 730
1994	wksw941-wksw948	do i=731 to 783
1995	wksw951-wksw958	do i=784 to 835
1996	wksw961-wksw968	do i=836 to 887
1997	wksw971-wksw978	do i=888 to 939
1998	wksw981-wksw988	do i=940 to 991
1999	wksw991-wksw998	do i=992 to 1043
2000	wksw001-wksw008	do i=1044 to 1096
2001	wksw011-wksw018	do i=1097 to 1148
2002	wksw021-wksw028	do i=1149 to 1200 **/

end;

```

do i=1 to 8;
if starfl=1 and stopw>1 and r5wks80>0 then do;
   r5wks80=-3; end;

```

```

if starfl=1 and stopw>52 and r5wks81>0 then do;
    r5wks81=-3; end;
if starfl=1 and r5wks82>0 and stopw>104 then do;
    r5wks82=-3; end;
if starfl=1 and r5wks83>0 and stopw>156 then do;
    r5wks83=-3; end;
if starfl=1 and r5wks84>0 and stopw>209 then do;
    r5wks84=-3; end;
if starfl=1 and r5wks85>0 and stopw>261 then do;
    r5wks85=-3; end;
if starfl=1 and r5wks86>0 and stopw>313 then do;
    r5wks86=-3; end;
if starfl=1 and r5wks87>0 and stopw>365 then do;
    r5wks87=-3; end;
if starfl=1 and r5wks88>0 and stopw>417 then do;
    r5wks88=-3; end;
if starfl=1 and r5wks89>0 and stopw>470 then do;
    r5wks89=-3; end;
if starfl=1 and r5wks90>0 and stopw>522 then do;
    r5wks90=-3; end;
if starfl=1 and r5wks91>0 and stopw>574 then do;
    r5wks91=-3; end;
if starfl=1 and r5wks92>0 and stopw>626 then do;
    r5wks92=-3; end;
if starfl=1 and r5wks93>0 and stopw>678 then do;
    r5wks93=-3; end;
if starfl=1 and r5wks94>0 and stopw>730 then do;
    r5wks94=-3; end;
if starfl=1 and r5wks95>0 and stopw>783 then do;
    r5wks95=-3; end;
if starfl=1 and r5wks96>0 and stopw>835 then do;
    r5wks96=-3; end;
if starfl=1 and r5wks97>0 and stopw>887 then do;
    r5wks97=-3; end;
if starfl=1 and r5wks98>0 and stopw>939 then do;
    r5wks98=-3; end;
if starfl=1 and r5wks99>0 and stopw>991 then do;
    r5wks99=-3; end;
if starfl=1 and r5wks00>0 and stopw>1043 then do;
    r5wks00=-3; end;
if starfl=1 and r5wks01>0 and stopw>1096 then do;
    r5wks01=-3; end;
if starfl=1 and r5wks02>0 and stopw>1148 then do;
    r5wks02=-3; end;

/* stop date invalid */
if stopfl=1 and r5wks80>0 and starw<53 then do;
    r5wks80=-3; end;
if stopfl=1 and r5wks81>0 and starw<105 then do;
    r5wks81=-3; end;
if stopfl=1 and r5wks82>0 and starw<157 then do;

```

```

r5wks82=-3; end;
if stopfl=1 and r5wks83>0 and starw<210 then do;
    r5wks83=-3; end;
if stopfl=1 and r5wks84>0 and starw<262 then do;
    r5wks84=-3; end;
if stopfl=1 and r5wks85>0 and starw<314 then do;
    r5wks85=-3; end;
if stopfl=1 and r5wks86>0 and starw<366 then do;
    r5wks86=-3; end;
if stopfl=1 and r5wks87>0 and starw<418 then do;
    r5wks87=-3; end;
if stopfl=1 and r5wks88>0 and starw<471 then do;
    r5wks88=-3; end;
if stopfl=1 and r5wks89>0 and starw<523 then do;
    r5wks89=-3; end;
if stopfl=1 and r5wks90>0 and starw<575 then do;
    r5wks90=-3; end;
if stopfl=1 and r5wks91>0 and starw<627 then do;
    r5wks91=-3; end;
if stopfl=1 and r5wks92>0 and starw<679 then do;
    r5wks92=-3; end;
if stopfl=1 and r5wks93>0 and starw<731 then do;
    r5wks93=-3; end;
if stopfl=1 and r5wks94>0 and starw<784 then do;
    r5wks94=-3; end;
if stopfl=1 and r5wks95>0 and starw<836 then do;
    r5wks95=-3; end;
if stopfl=1 and r5wks96>0 and starw<888 then do;
    r5wks96=-3; end;
if stopfl=1 and r5wks97>0 and starw<940 then do;
    r5wks97=-3; end;
if stopfl=1 and r5wks98>0 and starw<991 then do;
    r5wks98=-3; end;
if stopfl=1 and r5wks99>0 and starw<1044 then do;
    r5wks99=-3; end;
if stopfl=1 and r5wks00>0 and starw<1097 then do;
    r5wks00=-3; end;
if stopfl=1 and r5wks01>0 and starw<1149 then do;
    r5wks01=-3; end;
if stopfl=1 and r5wks02>0 and starw<1201 then do;
    r5wks02=-3; end;
end;

```

```

do i=1 to 8;
  if e200=-5 then do;
    r5wks80=-5;
    r5wks81=-5;
    /* and so on through */
    r5wks99=-5;
    r5wks00=-5;
    r5wks01=-5;

```

```
r5wks02=-5;  
end;  
end;
```

## **/\*Section 2: Merge in the information from previous rounds\*/**

```
/* Round 4 variable */  
array r4wks80 r4wks801-r4wks809;  
array r4wks81 r4wks811-r4wks819;  
    /* and so on through */  
array r4wks01 r4wks011-r4wks019;
```

```
/* Total created variable */  
array twks80 twks801-twks809;  
array twks81 twks811-twks819;  
    /* and so on through */  
array twks01 twks011-twks019;
```

```
do over twks80;  
    r4wks80=twks80;  
    r4wks81=twks81;  
        /* and so on through */  
    r4wks01=twks01;  
end;
```

/\* WKS945 is the Round 2 created variable for 5th job (i.e. 5th UID code) in 1994. WKS945 is the Round 3 created variable of the same description. r1wks945 is the variable that combines these two created variables (of the same description). Begin by renaming them to more familiar "r1wks, r2wks" etc. "twks" remains the total created variable. \*/

```
/* Note there are 7 jobs in Round 1, array matched to make programming easier */  
array r1uid (i) r1uid1-r1uid9;  
array r2uid (i) r2uid1-r2uid9;  
array r3uid (i) r3uid1-r3uid9;  
array r4uid (i) r4uid1-r4uid9;  
array r5uid (i) uid1-uid9;
```

```
/* Arrange the jobs by year for Round 1 */  
array r1wks80 (i) r1wks801-r1wks809;  
array r1wks81 (i) r1wks811-r1wks819;  
    /* and so on through */  
array r1wks99 (i) r1wks991-r1wks999;
```

```
/* Arrange the jobs by year for Round 2 */  
array r2wks80 (i) r2wks801-r2wks809;  
array r2wks81 (i) r2wks811-r2wks819;  
    /* and so on through */  
array r2wks00 (i) r2wks001-r2wks009;
```

```
/* Arrange the jobs by year for Round 3 */  
array r3wks80 (i) r3wks801-r3wks809;
```

```

array r3wks81 (i) r3wks811-r3wks819;
/* and so on through */
array r3wks00 (i) r3wks001-r3wks009;

/* Arrange the jobs by year for Round 4 */
array r4wks80 (i) r4wks801-r4wks809;
array r4wks81 (i) r4wks811-r4wks819;
/* and so on through */
array r4wks01 (i) r4wks011-r4wks019;

/* Arrange the jobs by year for Round 5 */
array r5wks80 (i) r5wks801-r5wks808;
array r5wks81 (i) r5wks811-r5wks818;
/* and so on through */
array r5wks02 (i) r5wks021-r5wks028;

/* Arrange the jobs by year for total c.v. */
array twks80 (i) twks801-twks808;
array twks81 (i) twks811-twks818;
/* and so on through */
array twks02 (i) twks021-twks028;

/* Initialize total weeks count to zero */
do i=1 to 8;
twks80=0; twks81=0; twks82=0; twks83=0; twks84=0; twks85=0; twks86=0; twks87=0; twks88=0;
twks89=0; twks90=0; twks91=0; twks92=0; twks93=0; twks94=0; twks95=0; twks96=0; twks97=0;
twks98=0; twks99=0; twks00=0; twks01=0; twks02=0;
end;

/* Accounts for Round 5 non-interview cases */
do i=1 to 8;
if e200=-5 then do;
twks80=-5; twks81=-5; twks82=-5; twks83=-5; twks84=-5; twks85=-5; twks86=-5; twks87=-5;
twks88=-5; twks89=-5; twks90=-5; twks91=-5; twks92=-5; twks93=-5; twks94=-5; twks95=-5;
twks96=-5; twks97=-5; twks98=-5; twks99=-5; twks00=-5; twks01=-5; twks02=-5;
end;
end;

/* Begin by updating the total created variable if the Round 5 created variable is positive. For this
variable, we are only worried about Round 5 jobs and previous round jobs that were worked in Round 5.
If a invalid negative value is given (don't know, refusal, or non-interview) then total created variable
will be that negative value. */

/* 1980 */
do i=1 to 8;
if r5wks80=>0 then do; twks80=r5wks80; end;
if -4<r5wks80<0 then do; twks80=-3; end;
end;

/* repeat above for 1981-1996 */

```

/\* Since 1997-2002 are the interview years, must account for both Round 5 and dli created variables.  
Begin with negative conditions for the Round 5 c.v. \*/

```
/* 1997 */
do i=1 to 8;
  if -4<r5wks97<0 then do; twks97=-3; end;
  if r5wks97=>0 then do; twks97=r5wks97; end;
end;

/* 1998 */
do i=1 to 8;
  if -4<r5wks98<0 then do; twks98=-3; end;
  if r5wks98=>0 then do; twks98=r5wks98; end;
end;

/* 1999 */
do i=1 to 8;
  if -4<r5wks99<0 then do; twks99=-3; end;
  if r5wks99=>0 then do; twks99=r5wks99; end;
end;

/* 2000 */
do i=1 to 8;
  if -4<r5wks00<0 then do; twks00=-3; end;
  if r5wks00>0 then do; twks00=r5wks00; end;
end;

/* 2001 */
do i=1 to 8;
  if -4<r5wks01<0 then do; twks01=-3; end;
  if r5wks01>0 then do; twks01=r5wks01; end;
end;

/* 2002 */
do i=1 to 8;
  if -4<r5wks02<0 then do; twks02=-3; end;
  if r5wks02>0 then do; twks02=r5wks02; end;
end;

/* When matching jobs from a dli Round to Round 5, we only need to worry about jobs in Round 5 with
a UID not beginning with "2002" since only those jobs that could have been worked in both rounds. */

array match1 (i) match11-match18;
array match2 (i) match21-match28;
array match3 (i) match31-match38;
array match4 (i) match41-match48;
array match5 (i) match51-match58;
array match6 (i) match61-match68;
array match7 (i) match71-match78;
array match8 (i) match81-match88;
array match9 (i) match91-match98;
```

```

do i=1 to 8;
  match1=0; match2=0; match3=0; match4=0; match5=0; match6=0; match7=0; match8=0; match9=0;
end;

/* Define match14 as the dummy variable that equals one when the first job on the Round1 UID roster
and the fourth job on the Round2 roster have the same UID. */

/* If any UID from the first position in dli Round matches with any UID in Round 5 */
do over r5uid;
  if r4uid1>0 and r5uid>0 and r4int>0 and intwk>0 then do;
    if r4uid1=r5uid then do; match1=1; end;
  end;
  if r3uid1>0 and r5uid>0 and r4int=-5 and intwk>0 then do;
    if r3uid1=r5uid then do; match1=1; end;
  end;
  if r2uid1>0 and r5uid>0 and r4int=-5 and r3int=-5 and intwk>0 then do;
    if r2uid1=r5uid then do; match1=1; end;
  end;
  if r1uid1>0 and r5uid>0 and r4int=-5 and r3int=-5 and r2int=-5 and intwk>0 then do;
    if r1uid1=r5uid then do; match1=1; end;
  end;
end;

/* If any UID from the second position in dli Round matches with any UID in Round 5 */
do over r5uid;
  if r4uid2>0 and r5uid>0 and r4int>0 and intwk>0 then do;
    if r4uid2=r5uid then do; match2=1; end;
  end;
  if r3uid2>0 and r5uid>0 and r4int=-5 and r3int>0 and intwk>0 then do;
    if r3uid2=r5uid then do; match2=1; end;
  end;
  if r2uid2>0 and r5uid>0 and r4int=-5 and r3int=-5 and intwk>0 then do;
    if r2uid2=r5uid then do; match2=1; end;
  end;
  if r1uid2>0 and r5uid>0 and r3int=-5 and r2int=-5 and intwk>0 then do;
    if r1uid2=r5uid then do; match2=1; end;
  end;
end;

/* If any UID from the third position in dli Round matches with any UID in Round 5 */
do over r5uid;
  if r4uid3>0 and r5uid>0 and r4int>0 and intwk>0 then do;
    if r4uid3=r5uid then do; match3=1; end;
  end;
  if r3uid3>0 and r5uid>0 and r4int=-5 and r3int>0 and intwk>0 then do;
    if r3uid3=r5uid then do; match3=1; end;
  end;
  if r2uid3>0 and r5uid>0 and r4int=-5 and r3int=-5 and intwk>0 then do;
    if r2uid3=r5uid then do; match3=1; end;
  end;

```

```

if r1uid3>0 and r5uid>0 and r4int=-5 and r3int=-5 and r2int=-5 and intwk>0 then do;
    if r1uid3=r5uid then do; match3=1; end;
end;
end;

/* If any UID from the fourth position in dli Round matches with any UID in Round 5 */
do over r5uid;
    if r4uid4>0 and r5uid>0 and r4int>0 and intwk>0 then do;
        if r4uid4=r5uid then do; match4=1; end;
    end;
    if r3uid4>0 and r5uid>0 and r4int=-5 and r3int>0 and intwk>0 then do;
        if r3uid4=r5uid then do; match4=1; end;
    end;
    if r2uid4>0 and r5uid>0 and r4int=-5 and r3int=-5 and intwk>0 then do;
        if r2uid4=r5uid then do; match4=1; end;
    end;
    if r1uid4>0 and r5uid>0 and r4int=-5 and r3int=-5 and r2int=-5 and intwk>0 then do;
        if r1uid4=r5uid then do; match4=1; end;
    end;
end;

/* If any UID from the fifth position in dli Round matches with any UID in Round 5 */
do over r5uid;
    if r4uid5>0 and r5uid>0 and r4int>0 and intwk>0 then do;
        if r4uid5=r5uid then do; match5=1; end;
    end;
    if r3uid5>0 and r5uid>0 and r4int=-5 and r3int>0 and intwk>0 then do;
        if r3uid5=r5uid then do; match5=1; end;
    end;
    if r2uid5>0 and r5uid>0 and r4int=-5 and r3int=-5 and intwk>0 then do;
        if r2uid5=r5uid then do; match5=1; end;
    end;
    if r1uid5>0 and r5uid>0 and r4int=-5 and r3int=-5 and r2int=-5 and intwk>0 then do;
        if r1uid5=r5uid then do; match5=1; end;
    end;
end;

/* If any UID from the sixth position in dli Round matches with any UID in Round 5 */
do over r5uid;
    if r4uid6>0 and r5uid>0 and r4int>0 and intwk>0 then do;
        if r4uid6=r5uid then do; match6=1; end;
    end;
    if r3uid6>0 and r5uid>0 and r4int=-5 and r3int>0 and intwk>0 then do;
        if r3uid6=r5uid then do; match6=1; end;
    end;
    if r2uid6>0 and r5uid>0 and r4int=-5 and r3int=-5 and intwk>0 then do;
        if r2uid6=r5uid then do; match6=1; end;
    end;
    if r1uid6>0 and r5uid>0 and r4int=-5 and r3int=-5 and r2int=-5 and intwk>0 then do;
        if r1uid6=r5uid then do; match6=1; end;
    end;

```

```

end;

/* If any UID from the seventh position in dli Round matches with any UID in Round 5 */
do over r5uid;
  if r4uid7>0 and r5uid>0 and r4int>0 and intwk>0 then do;
    if r4uid7=r5uid then do; match7=1; end;
  end;
  if r3uid7>0 and r5uid>0 and r4int=-5 and r3int>0 and intwk>0 then do;
    if r3uid7=r5uid then do; match7=1; end;
  end;
  if r2uid7>0 and r5uid>0 and r4int=-5 and r3int=-5 and intwk>0 then do;
    if r2uid7=r5uid then do; match7=1; end;
  end;
  if r1uid7>0 and r5uid>0 and r4int=-5 and r3int=-5 and r2int=-5 and intwk>0 then do;
    if r1uid7=r5uid then do; match7=1; end;
  end;
end;

/* If any UID from the eighth position in dli Round matches with any UID in Round 5 */
do over r5uid;
  if r4uid8>0 and r5uid>0 and r4int>0 and intwk>0 then do;
    if r4uid8=r5uid then do; match8=1; end;
  end;
  if r3uid8>0 and r5uid>0 and r4int=-5 and r3int>0 and intwk>0 then do;
    if r3uid8=r5uid then do; match8=1; end;
  end;
  if r2uid8>0 and r5uid>0 and r4int=-5 and r3int=-5 and intwk>0 then do;
    if r2uid8=r5uid then do; match8=1; end;
  end;
end;

/* If any UID from the ninth position in dli Round matches with any UID in Round 5 */
do over r5uid;
  if r4uid9>0 and r5uid>0 and r4int>0 and intwk>0 then do;
    if r4uid9=r5uid then do; match9=1; end;
  end;
  if r3uid9>0 and r5uid>0 and r4int=-5 and r3int>0 and intwk>0 then do;
    if r3uid9=r5uid then do; match9=1; end;
  end;
  if r2uid9>0 and r5uid>0 and r4int=-5 and r3int=-5 and intwk>0 then do;
    if r2uid9=r5uid then do; match9=1; end;
  end;
end;

/* Now reassign total created variable if there is an identical job was worked in both rounds. Using the
match variable, a job worked in both rounds will update the total created variable. Need to add positivity
condition on the weeks worked values since some of the values equal -3. */

/* For respondents last interviewed in Round 4, 2000 */
if r4int_y=2001 then do;
  do over match1;

```

```

if match1=1 then do;
  if r4wks801 not in (-5,-4,0,.) then twks80=r4wks801;
  if r4wks811 not in (-5,-4,0,.) then twks81=r4wks811;
    /* and so on through */
  if r4wks001 not in (-5,-4,0,.) then twks00=r4wks001;
end;
end;

do over match1;
  if match1=1 & r5wks01=>0 & r4wks011=>0 then do;
    twks01=r4wks011+r5wks01;
  end;
  if match1=1 & (-4<r5wks01<0 or -4<r4wks011<0) then do;
    twks01=-3;
  end;
end;

```

/\*At this point the program repeats the above code for match2-match9. Contact NLS User Services for details.\*/

```

/* For respondents last interviewed in Round 4, 2000 */
if r4int_y=2000 then do;
  do over match1;
    if match1=1 then do;
      if r4wks801 not in (-5,-4,0,.) then twks80=r4wks801;
      if r4wks811 not in (-5,-4,0,.) then twks81=r4wks811;
        /* and so on through */
      if r4wks991 not in (-5,-4,0,.) then twks99=r4wks991;
    end;
  end;
end;

do over match1;
  if match1=1 and r5wks00=>0 and r4wks001=>0 then do;
    twks00=r4wks001+r5wks00;
  end;
  if match1=1 and (-4<r5wks00<0 or -4<r4wks001<0) then do;
    twks00=-3;
  end;
end;

```

/\*At this point the program repeats the above code for match2-match9. Contact NLS User Services for details.\*/

```

/* For respondents last interviewed in Round 3, 2000 */
if r3int_y=2000 and r4int=-5 then do;
  do over match1;
    if match1=1 then do;
      if r3wks801 not in (-5,-4,0,.) then twks80=r3wks801;
      if r3wks811 not in (-5,-4,0,.) then twks81=r3wks811;
        /* and so on through */
      if r3wks991 not in (-5,-4,0,.) then twks99=r3wks991;
    end;
  end;

```

```

end;
end;

do over match1;
  if match1=1 and r5wks00=>0 and r3wks001=>0 then do;
    twks00=r3wks001+r5wks00;
  end;
  if match1=1 and (-4<r5wks00<0 or -4<r3wks001<0) then do;
    twks00=-3;
  end;
end;

```

/\*At this point the program repeats the above code for match2-match9. Contact NLS User Services for details.\*/

/\* For respondents last interviewed in Round 3, 1999 \*/

```

if r3int_y=1999 and r4int=-5 then do;
  do over match1;
    if match1=1 then do;
      if r3wks801 not in (-5,-4,0,.) then twks80=r3wks801;
      if r3wks811 not in (-5,-4,0,.) then twks81=r3wks811;
      /* and so on through */
      if r3wks981 not in (-5,-4,0,.) then twks98=r3wks981;
    end;
  end;

```

```

do over match1;
  if match1=1 and r5wks99=>0 and r3wks991=>0 then do;
    twks99=r3wks991+r5wks99;
  end;
  if match1=1 and (-4<r5wks99<0 or -4<r3wks991<0) then do;
    twks99=-3;
  end;
end;

```

/\*At this point the program repeats the above code for match2-match9. Contact NLS User Services for details.\*/

/\* For respondents last interviewed in Round 2, 1999 \*/

```

if r2int_y=1999 and r3int=-5 and r4int=-5 then do;
  do over match1;
    if match1=1 then do;
      if r2wks801 not in (-5,-4,0,.) then twks80=r2wks801;
      if r2wks811 not in (-5,-4,0,.) then twks81=r2wks811;
      /* and so on through */
      if r2wks981 not in (-5,-4,0,.) then twks98=r2wks981;
    end;
  end;

```

```

do over match1;
  if match1=1 and r5wks99=>0 and r2wks991=>0 then do;

```

```

twks99=r2wks991+r5wks99;
end;
if match1=1 and (-4<r5wks99<0 or -4<r2wks991<0) then do;
  twks99=-3;
end;
end;

/* For respondents last interviewed in Round 2, 1998 */
if r2int_y=1998 and r3int=-5 and r4int=-5 then do;
  do over match1;
    if match1=1 then do;
      if r2wks801 not in (-5,-4,0,.) then twks80=r2wks801;
      if r2wks811 not in (-5,-4,0,.) then twks81=r2wks811;
      /* and so on through */
      if r2wks971 not in (-5,-4,0,.) then twks97=r2wks971;
    end;
  end;
end;

do over match1;
  if match1=1 and r5wks98=>0 and r2wks981=>0 then do;
    twks98=r2wks981+r5wks98;
  end;
  if match1=1 and (-4<r5wks98<0 or -4<r2wks981<0) then do;
    twks98=-3;
  end;
end;
end;

/* For respondents last interviewed in Round 1, 1998 */
if r1int_y=1998 and r2int=-5 and r3int=-5 and r4int=-5 and intwk>0 then do;
  do over match1;
    if match1=1 then do;
      if r1wks801 not in (-5,-4,0,.) then twks80=r1wks801;
      if r1wks811 not in (-5,-4,0,.) then twks81=r1wks811;
      /* and so on through */
      if r1wks971 not in (-5,-4,0,.) then twks97=r1wks971;
    end;
  end;
end;

do over match1;
  if match1=1 and r5wks98=>0 and r1wks981=>0 then do;
    twks98=r1wks981+r5wks98;
  end;
  if match1=1 and (-4<r5wks98<0 or -4<r1wks981<0) then do;
    twks98=-3;
  end;
end;
end;

/* For respondents last interviewed in Round 1, 1997 */
if r1int_y=1997 and r2int=-5 and r3int=-5 and r4int=-5 and intwk>0 then do;
  do over match1;
    if match1=1 then do;

```

```

if r1wks801 not in (-5,-4,0,.) then twks80=r1wks801;
if r1wks811 not in (-5,-4,0,.) then twks81=r1wks811;
/* and so on through */
if r1wks961 not in (-5,-4,0,.) then twks96=r1wks961;
end;
end;

do over match1;
  if match1=1 and r5wks97=>0 and r1wks971=>0 then do;
    twks97=r1wks971+r5wks97;
    end;
  if match1=1 and (-4<r5wks97<0 or -4<r1wks971<0) then do;
    twks97=-3;
    end;
  end;
end;

endsas;

```

---

## Total Tenure at Job #x as of the Survey Date

**Variables Created:** CV\_WKSWK\_JOB\_DLI.01 - CV\_WKSWK\_JOB\_DLI.09

### Programs Used

This program uses [emp\\_begin.sas](#) as input (see the first page of this appendix for details).

This program creates a variable for each job calculating the total length of job tenure in weeks (excluding within-job gaps) since the respondent's 14<sup>th</sup> birthday. A variable is created for each potential job even if the respondent has no data for that job, with the default value set to zero (0). The most jobs held by any respondent in round 5 was eight, so variables are created for eight jobs for each respondent.

---

### /\* Section 1: Create a variable for each potential job (8) relating the total length of job tenure in weeks excluding within-job gaps.\*/

```

array starw (i) starw1-starw8;
array job1wks (i) wk1_1-wk1_1200;
array job2wks (i) wk2_1-wk2_1200;
array job3wks (i) wk3_1-wk3_1200;
array job4wks (i) wk4_1-wk4_1200;
array job5wks (i) wk5_1-wk5_1200;
array job6wks (i) wk6_1-wk6_1200;
array job7wks (i) wk7_1-wk7_1200;
array job8wks (i) wk8_1-wk8_1200;
array starfl (i) starfl_1-starfl_8;

```

```

array stopfl (i) stopfl_1-stopfl_8;

** Calculate cumulative weeks on individual jobs for all years **;

r5ten1=0;          r5ten2=0;
r5ten3=0;          r5ten4=0;
r5ten5=0;          r5ten6=0;
r5ten7=0;          r5ten8=0;

do i=1 to 1200;
  if job1wks=1 then do; r5ten1=r5ten1+1; end;
  if job2wks=1 then do; r5ten2=r5ten2+1; end;
    /* and so on through */
  if job8wks=1 then do; r5ten8=r5ten8+1; end;
end;

flag=0;

do i=1 to 1200;
  if job1wks=-3 then do; r5ten1=-3; flag=1; end;
  if job2wks=-3 then do; r5ten2=-3; flag=2; end;
    /* and so on through */
  if job8wks=-3 then do; r5ten8=-3; flag=8;
  end;
end;

do i=1 to 1200;
  if starfl_1=1 then do; r5ten1=-3; end;
  if starfl_2=1 then do; r5ten2=-3; end;
    /* and so on through */
  if starfl_8=1 then do; r5ten8=-3; end;

  if stopfl_1=1 then do; r5ten1=-3; end;
  if stopfl_2=1 then do; r5ten2=-3; end;
    /* and so on through */
  if stopfl_8=1 then do; r5ten8=-3; end;
end;

if e200=-5 then do;
  r5ten1=-5; r5ten2=-5; r5ten3=-5; r5ten4=-5; r5ten5=-5; r5ten6=-5; r5ten7=-5; r5ten8=-5;
end;

array r4ten (i) r4ten1-r4ten9;
array totten (i) totten1-totten9;

do over totten; r4ten=totten; end;

/* Section 2: Match to jobs in previous rounds */

/* Begin by renaming the infile variables */
array r1ten (i) r1ten1-r1ten9; /* New R1 c.v. name */

```

```

array r2ten (i) r2ten1-r2ten9; /* New R2 c.v. name */
array r3ten (i) r3ten1-r3ten9; /* New R3 c.v. name */
array r4ten (i) r4ten1-r4ten9; /* New R4 c.v. name */
array r5ten (i) r5ten1-r5ten9; /* New R4 c.v. name */
array totten (i) totten1-totten8; /* Total c.v. name */

do i=1 to 8; totten=0; end;

/* When matching jobs from a dli Round to Round 5, we only need to worry about jobs in Round 5 with
a UID not beginning with "2002" since only those jobs that could have been worked in both rounds. */

array match1 (i) match11-match18;
array match2 (i) match21-match28;
      /* and so on through */
array match9 (i) match91-match98;

do i=1 to 8;
  match1=0; match2=0; match3=0; match4=0; match5=0; match6=0; match7=0; match8=0; match9=0;
end;

/* Define match14 as the dummy variable that equals one when the first job on the dli UID roster and
the fourth job on the Round2 roster have the same UID. */

/* Note there are 7 jobs in Round 1, array matched to make programming easier */
array r1uid (i) r1uid1-r1uid9;
array r2uid (i) r2uid1-r2uid9;
array r3uid (i) r3uid1-r3uid9;
array r4uid (i) r4uid1-r4uid9;
array r5uid (i) uid1-uid9;

/* If any UID from the first position in dli Round matches with any UID in Round 5 */
do over r5uid;
  if r4uid1>0 and r5uid>0 and r4int>0 and intwk>0 then do;
    if r4uid1=r5uid then do; match1=1; end;
  end;
  if r3uid1>0 and r5uid>0 and r4int=-5 and intwk>0 then do;
    if r3uid1=r5uid then do; match1=1; end;
  end;
  if r2uid1>0 and r5uid>0 and r4int=-5 and r3int=-5 and intwk>0 then do;
    if r2uid1=r5uid then do; match1=1; end;
  end;
  if r1uid1>0 and r5uid>0 and r4int=-5 and r3int=-5 and r2int=-5 and intwk>0 then do;
    if r1uid1=r5uid then do; match1=1; end;
  end;
end;

/* If any UID from the second position in dli Round matches with any UID in Round 5 */
do over r5uid;
  if r4uid2>0 and r5uid>0 and r4int>0 and intwk>0 then do;
    if r4uid2=r5uid then do; match2=1; end;
  end;

```

```

if r3uid2>0 and r5uid>0 and r4int=-5 and r3int>0 and intwk>0 then do;
    if r3uid2=r5uid then do; match2=1; end;
end;
if r2uid2>0 and r5uid>0 and r4int=-5 and r3int=-5 and intwk>0 then do;
    if r2uid2=r5uid then do; match2=1; end;
end;
if r1uid2>0 and r5uid>0 and r4int=-5 and r3int=-5 and r2int=-5 and intwk>0 then do;
    if r1uid2=r5uid then do; match2=1; end;
end;
end;

```

```

/* If any UID from the third position in dli Round matches with any UID in Round 5 */
do over r5uid;
    if r4uid3>0 and r5uid>0 and r4int>0 and intwk>0 then do;
        if r4uid3=r5uid then do; match3=1; end;
    end;
    if r3uid3>0 and r5uid>0 and r4int=-5 and r3int>0 and intwk>0 then do;
        if r3uid3=r5uid then do; match3=1; end;
    end;
    if r2uid3>0 and r5uid>0 and r4int=-5 and r3int=-5 and intwk>0 then do;
        if r2uid3=r5uid then do; match3=1; end;
    end;
    if r1uid3>0 and r5uid>0 and r4int=-5 and r3int=-5 and r2int=-5 and intwk>0 then do;
        if r1uid3=r5uid then do; match3=1; end;
    end;
end;

```

```

/* If any UID from the fourth position in dli Round matches with any UID in Round 5 */
do over r5uid;
    if r4uid4>0 and r5uid>0 and r4int>0 and intwk>0 then do;
        if r4uid4=r5uid then do; match4=1; end;
    end;
    if r3uid4>0 and r5uid>0 and r4int=-5 and r3int>0 and intwk>0 then do;
        if r3uid4=r5uid then do; match4=1; end;
    end;
    if r2uid4>0 and r5uid>0 and r4int=-5 and r3int=-5 and intwk>0 then do;
        if r2uid4=r5uid then do; match4=1; end;
    end;
    if r1uid4>0 and r5uid>0 and r4int=-5 and r3int=-5 and r2int=-5 and intwk>0 then do;
        if r1uid4=r5uid then do; match4=1; end;
    end;
end;

```

```

/* If any UID from the fifth position in dli Round matches with any UID in Round 5 */
do over r5uid;
    if r4uid5>0 and r5uid>0 and r4int>0 and intwk>0 then do;
        if r4uid5=r5uid then do; match5=1; end;
    end;
    if r3uid5>0 and r5uid>0 and r4int=-5 and r3int>0 and intwk>0 then do;
        if r3uid5=r5uid then do; match5=1; end;
    end;

```

```
if r2uid5>0 and r5uid>0 and r4int=-5 and r3int=-5 and intwk>0 then do;
    if r2uid5=r5uid then do; match5=1; end;
end;
if r1uid5>0 and r5uid>0 and r4int=-5 and r3int=-5 and r2int=-5 and intwk>0 then do;
    if r1uid5=r5uid then do; match5=1; end;
end;
end;
```

```
/* If any UID from the sixth position in dli Round matches with any UID in Round 5 */
do over r5uid;
    if r4uid6>0 and r5uid>0 and r4int>0 and intwk>0 then do;
        if r4uid6=r5uid then do; match6=1; end;
    end;
    if r3uid6>0 and r5uid>0 and r4int=-5 and r3int>0 and intwk>0 then do;
        if r3uid6=r5uid then do; match6=1; end;
    end;
    if r2uid6>0 and r5uid>0 and r4int=-5 and r3int=-5 and intwk>0 then do;
        if r2uid6=r5uid then do; match6=1; end;
    end;
    if r1uid6>0 and r5uid>0 and r4int=-5 and r3int=-5 and r2int=-5 and intwk>0 then do;
        if r1uid6=r5uid then do; match6=1; end;
    end;
end;
```

```
/* If any UID from the seventh position in dli Round matches with any UID in Round 5 */
do over r5uid;
    if r4uid7>0 and r5uid>0 and r4int>0 and intwk>0 then do;
        if r4uid7=r5uid then do; match7=1; end;
    end;
    if r3uid7>0 and r5uid>0 and r4int=-5 and r3int>0 and intwk>0 then do;
        if r3uid7=r5uid then do; match7=1; end;
    end;
    if r2uid7>0 and r5uid>0 and r4int=-5 and r3int=-5 and intwk>0 then do;
        if r2uid7=r5uid then do; match7=1; end;
    end;
    if r1uid7>0 and r5uid>0 and r4int=-5 and r3int=-5 and r2int=-5 and intwk>0 then do;
        if r1uid7=r5uid then do; match7=1; end;
    end;
end;
```

```
/* If any UID from the eighth position in dli Round matches with any UID in Round 5 */
do over r5uid;
    if r4uid8>0 and r5uid>0 and r4int>0 and intwk>0 then do;
        if r4uid8=r5uid then do; match8=1; end;
    end;
    if r3uid8>0 and r5uid>0 and r4int=-5 and r3int>0 and intwk>0 then do;
        if r3uid8=r5uid then do; match8=1; end;
    end;
    if r2uid8>0 and r5uid>0 and r4int=-5 and r3int=-5 and intwk>0 then do;
        if r2uid8=r5uid then do; match8=1; end;
    end;
```

```

end;

/* If any UID from the ninth position in dli Round matches with any UID in Round 5 */
do over r5uid;
  if r4uid9>0 and r5uid>0 and r4int>0 and intwk>0 then do;
    if r4uid9=r5uid then do; match9=1; end;
  end;
  if r3uid9>0 and r5uid>0 and r4int=-5 and r3int>0 and intwk>0 then do;
    if r3uid9=r5uid then do; match9=1; end;
  end;
  if r2uid9>0 and r5uid>0 and r4int=-5 and r3int=-5 and intwk>0 then do;
    if r2uid9=r5uid then do; match9=1; end;
  end;
end;

/* The only dli Round jobs that we are worried about are the one that were also worked in Round 5. */
do over totten; totten=r5ten; end;

/* Now reassign total tenure if there is a job was worked in both rounds. Using the match variable, a job
worked in both round 5 & the dli round will update the total tenure variable. Need to add positivity
condition on the tenure values since some of the values equal -3. */

do over match1;
  if match1=1 and r5ten=>0 and r4ten1=>0 then do;
    totten=r4ten1+r5ten; end;
  if match1=1 and -4<r4ten1<0 then do;
    totten=-3; end;
  if match1=1 and r5ten=>0 and r3ten1=>0 and r4int=-5 then do;
    totten=r3ten1+r5ten; end;
  if match1=1 and -4<r3ten1<0 and r4int=-5 then do;
    totten=-3; end;
  if match1=1 and r5ten=>0 and r2ten1=>0 and r4int=-5 and r3int=-5 then do;
    totten=r2ten1+r5ten; end;
  if match1=1 and -4<r2ten1<0 and r4int=-5 and r3int=-5 then do;
    totten=-3; end;
  if match1=1 and r5ten=>0 and r1ten1=>0 and r4int=-5 and r3int=-5 and r2int=-5 then do;
    totten=r1ten1+r5ten; end;
  if match1=1 and -4<r1ten1<0 and r4int=-5 and r3int=-5 and r2int=-5 then do;
    totten=-3; end;
end;

/*repeat the above code for match2-match9*/

endsas;

```

---

## Total Hours Worked in 19xx/20xx

**Variables Created:** CV\_HOURS\_WK\_YR\_ALL.80 - CV\_HOURS\_WK\_YR\_ALL.02

CV\_HOURS\_WK\_YR\_ET.80 - CV\_HOURS\_WK\_YR\_ET.02  
CV\_HOURS\_WK\_YR\_SE.98 - CV\_HOURS\_WK\_YR\_SE.02

## Programs Used

This program uses [emp\\_begin.sas](#) as input (see the first page of this appendix for details).

This program calculates the number of hours worked by the respondent at all jobs in each calendar year. Three sets of variables are created: one for all jobs, one for employee-type jobs, and one for self-employed jobs for respondents age 18 and older in rounds 4 and 5. See the [introduction to this appendix for important information about the universes for each of these variables](#). A variable is created for each respondent even if the respondent has worked no jobs in a given year with the default value set to zero (0). Note that when both "starting hours" and "current hours" are reported, the latter are used to construct these measures.

---

```
*** Organize hours for each job;
array starw (k) starw1-starw9;
array stopw (k) stopw1-stopw9;

/* shrs1 - starting hours           ehrs1 - ending hours */
array shrs (k) shrs1-shrs9;
array ehrs (k) ehrs1-ehrs9;
array hours (k) hours1-hours9;
array hrck (k) hrck1-hrck9;

array e23901 (k) e239011-e239018;           array e24501 (k) e245011-e245018;
array e34402 (k) e344021-e344028;           array e34403 (k) e344031-e344038;
array e34428 (k) e344281-e344288;           array e37904 (k) e379041-e379048;
array e38000f (k) e38000f1-e38000f8;          array e38103 (k) e381031-e381038;
array e38105 (k) e381051-e381058;           array e59901 (k) e599011-e599018;
array e88000 (k) e880001-e880008;           array e88501 (k) e885011-e885018;
array e98402 (k) e984021-e984208;           array e98403 (k) e984031-e984038;
array e98429 (k) e984291-e984298;           array e100231 (k) e1002311-e1002318;
array e100232 (k) e1002321-e1002328;          array e22610 (k) e226101-e226108;
array e22611 (k) e226111-e226118;           array starfl (k) starfl_1-starfl_8;

/* Define the number of hours per week worked at the start date and the end date of a job. The code
below defines the starting hours and ending hours per week at those jobs, including overtime. */

do k=1 to 8;
  if e23901>-4 then do; shrs=e23901; end;
  if e23901>-4 and e24501>-4 then do; shrs=e23901+e24501; end;
  if e34402>-4 then do; shrs=e34402; end;
  if e34403>-4 then do; shrs=e34403; end;
  if e34403>-4 and e34428>-4 then do; shrs=e34403+e34428; end;
  if e37904>-4 then do; ehrs=e37904; end;
  if e38000f>-4 then do; ehrs=e38000f; end;
  if e38103>-4 then do; ehrs=e38103; end;
  if e38103>-4 and e38105>-4 then do; ehrs=e38103+e38105; end;
```

```

if e59901>-4 then do; shrs=e59901; end;
if e88000>-4 then do; shrs=e88000; end;
if e88501>-4 and e88000>-4 then do; shrs=e88501+e88000; end;
if e98402>-4 then do; shrs=e98402; end;
if e98429>-4 then do; shrs=e98429; end;
if e98429>-4 and e98403>-4 then do; shrs=e98403+e98429; end;
if e22610>-4 then do; shrs=e22610; end;
if e22611>-4 then do; shrs=e22611; end;
if e22610>-4 and e22611>-4 then do; shrs=e22610+e22611; end;
if e100231>-4 then do; shrs=e100231; end;
if e100231>-4 then do; shrs=e100232; end;
if e100231>-4 and e100232>-4 then do; shrs=e100231+e100232; end;
end;

```

/\* The code below decides which hours per week total (starting or ending) will be used in determining hours worked per year. Hours per week starts with 0, is updated if starting hours are available, and is updated once more if ending hours are available. \*/

\* Set default hours to zero ;  
do over hours; hours=0; hrck=0; end;

\* Take starting hours if reported (eliminates -4's);  
do over shrs;  
if shrs=>0 then do; hours=shrs; end;  
if -4<shrs<0 then do; hrck=shrs; hours=shrs; end;  
end;

\* Write over if end hours reported;  
do over ehrs;  
if ehrs=>0 then do; hours=ehrs; end;  
if -4<ehrs<0 then do; hrck=ehrs; hours=ehrs; end;  
end;

array job1wks (i) wk1\_1-wk1\_1200;  
array job3wks (i) wk3\_1-wk3\_1200;  
array job5wks (i) wk5\_1-wk5\_1200;  
array job7wks (i) wk7\_1-wk7\_1200;

array job2wks (i) wk2\_1-wk2\_1200;  
array job4wks (i) wk4\_1-wk4\_1200;  
array job6wks (i) wk6\_1-wk6\_1200;  
array job8wks (i) wk8\_1-wk8\_1200;

\*\* Calculate cumulative weeks on individual jobs for each year \*\*;

/\* 1980 \*/  
array wks80 (i) wks801-wks808;  
array swks80 (i) swks801-swks808;  
array ewks80 (i) ewks801-ewks808;  
array ah80 (i) ah801-ah808;  
array sah80 (i) sah801-sah808;  
array eah80 (i) eah801-eah808;  
  
do i=1 to 8; wks80=0; swks80=0; ewks80=0; ah80=0; sah80=0; eah80=0; end;

/\* Update hours counter when respondent reports employment in weeks 1 to 52. \*/

```

do i=1 to 52;
  if job1wks=1 then do; wks801=wks801+1; end;
  if job2wks=1 then do; wks802=wks802+1; end;
  if job3wks=1 then do; wks803=wks803+1; end;
  if job4wks=1 then do; wks804=wks804+1; end;
  if job5wks=1 then do; wks805=wks805+1; end;
  if job6wks=1 then do; wks806=wks806+1; end;
  if job7wks=1 then do; wks807=wks807+1; end;
  if job8wks=1 then do; wks808=wks808+1; end;

/* Self-employed jobs */
  if job1wks=1 and self1=1 then do; swks801=swks801+1; end;
  if job2wks=1 and self2=1 then do; swks802=swks802+1; end;
  if job3wks=1 and self3=1 then do; swks803=swks803+1; end;
  if job4wks=1 and self4=1 then do; swks804=swks804+1; end;
  if job5wks=1 and self5=1 then do; swks805=swks805+1; end;
  if job6wks=1 and self6=1 then do; swks806=swks806+1; end;
  if job7wks=1 and self7=1 then do; swks807=swks807+1; end;
  if job8wks=1 and self8=1 then do; swks808=swks808+1; end;

/* Employee-type jobs */
  if job1wks=1 and self1=0 then do; ewks801=ewks801+1; end;
  if job2wks=1 and self2=0 then do; ewks802=ewks802+1; end;
  if job3wks=1 and self3=0 then do; ewks803=ewks803+1; end;
  if job4wks=1 and self4=0 then do; ewks804=ewks804+1; end;
  if job5wks=1 and self5=0 then do; ewks805=ewks805+1; end;
  if job6wks=1 and self6=0 then do; ewks806=ewks806+1; end;
  if job7wks=1 and self7=0 then do; ewks807=ewks807+1; end;
  if job8wks=1 and self8=0 then do; ewks808=ewks808+1; end;
end;

do i=1 to 52;
  if hours1>0 and job1wks ne -3 then ah801=hours1*wks801;
  if hours2>0 and job2wks ne -3 then ah802=hours2*wks802;
  if hours3>0 and job3wks ne -3 then ah803=hours3*wks803;
  if hours4>0 and job4wks ne -3 then ah804=hours4*wks804;
  if hours5>0 and job5wks ne -3 then ah805=hours5*wks805;
  if hours6>0 and job6wks ne -3 then ah806=hours6*wks806;
  if hours7>0 and job7wks ne -3 then ah807=hours7*wks807;
  if hours8>0 and job8wks ne -3 then ah808=hours8*wks808;

  if hours1>0 and job1wks ne -3 and self1=1 then sah801=hours1*swks801; /* 1-7-02 Self-employed
jobs */
  if hours2>0 and job2wks ne -3 and self2=1 then sah802=hours2*swks802;
  if hours3>0 and job3wks ne -3 and self3=1 then sah803=hours3*swks803;
  if hours4>0 and job4wks ne -3 and self4=1 then sah804=hours4*swks804;
  if hours5>0 and job5wks ne -3 and self5=1 then sah805=hours5*swks805;
  if hours6>0 and job6wks ne -3 and self6=1 then sah806=hours6*swks806;
  if hours7>0 and job7wks ne -3 and self7=1 then sah807=hours7*swks807;
  if hours8>0 and job8wks ne -3 and self8=1 then sah808=hours8*swks808;

```

```

if hours1>0 and job1wks ne -3 and self1=0 then eah801=hours1*ewks801; /* 1-7-02 Employee-type
jobs */
if hours2>0 and job2wks ne -3 and self2=0 then eah802=hours2*ewks802;
if hours3>0 and job3wks ne -3 and self3=0 then eah803=hours3*ewks803;
if hours4>0 and job4wks ne -3 and self4=0 then eah804=hours4*ewks804;
if hours5>0 and job5wks ne -3 and self5=0 then eah805=hours5*ewks805;
if hours6>0 and job6wks ne -3 and self6=0 then eah806=hours6*ewks806;
if hours7>0 and job7wks ne -3 and self7=0 then eah807=hours7*ewks807;
if hours8>0 and job8wks ne -3 and self8=0 then eah808=hours8*ewks808;
end;

```

```

do i=1 to 52; /* Define negative values for each job separately. */
  if job1wks=-3 or 0<hours1<-4 then ah801=-3;
  if (job1wks=-3 or 0<hours1<-4) and self1=1 then sah801=-3;
  if (job1wks=-3 or 0<hours1<-4) and self1=0 then eah801=-3;
  if job2wks=-3 or 0<hours2<-4 then ah802=-3;
  if (job2wks=-3 or 0<hours2<-4) and self2=1 then sah802=-3;
  if (job2wks=-3 or 0<hours2<-4) and self2=0 then eah802=-3;
  if job3wks=-3 or 0<hours3<-4 then ah803=-3;
  if (job3wks=-3 or 0<hours3<-4) and self3=1 then sah803=-3;
  if (job3wks=-3 or 0<hours3<-4) and self3=0 then eah803=-3;
  if job4wks=-3 or 0<hours4<-4 then ah804=-3;
  if (job4wks=-3 or 0<hours4<-4) and self4=1 then sah804=-3;
  if (job4wks=-3 or 0<hours4<-4) and self4=0 then eah804=-3;
  if job5wks=-3 or 0<hours5<-4 then ah805=-3;
  if (job5wks=-3 or 0<hours5<-4) and self5=1 then sah805=-3;
  if (job5wks=-3 or 0<hours5<-4) and self5=0 then eah805=-3;
  if job6wks=-3 or 0<hours6<-4 then ah806=-3;
  if (job6wks=-3 or 0<hours6<-4) and self6=1 then sah806=-3;
  if (job6wks=-3 or 0<hours6<-4) and self6=0 then eah806=-3;
  if job7wks=-3 or 0<hours7<-4 then ah807=-3;
  if (job7wks=-3 or 0<hours7<-4) and self7=1 then sah807=-3;
  if (job7wks=-3 or 0<hours7<-4) and self7=0 then eah807=-3;
  if job8wks=-3 or 0<hours8<-4 then ah808=-3;
  if (job8wks=-3 or 0<hours8<-4) and self8=1 then sah808=-3;
  if (job8wks=-3 or 0<hours8<-4) and self8=0 then eah808=-3;
end;

```

/\* Sum up the valid hours of each job \*/

```

if ah801=>0 and ah802=>0 and ah803=>0 and ah804=>0 and ah805=>0 and ah806=>0 and ah807=>0
and
ah808=>0 then do;
  r5hrs80=ah801+ah802+ah803+ah804+ah805+ah806+ah807+ah808;
end;
if sah801=>0 and sah802=>0 and sah803=>0 and sah804=>0 and sah805=>0 and sah806=>0 and
sah807=>0 and
sah808=>0 then do;
  r5shrs80=sah801+sah802+sah803+sah804+sah805+sah806+sah807+sah808;
end;
if eah801=>0 and eah802=>0 and eah803=>0 and eah804=>0 and eah805=>0 and eah806=>0 and
eah807=>0 and

```

```

eah808=>0 then do;
  r5ehrs80=eah801+eah802+eah803+eah804+eah805+eah806+eah807+eah808;
end;

/* Place in invalid skips */
if ah801=-3 or ah802=-3 or ah803=-3 or ah804=-3 or ah805=-3 or ah806=-3 or ah807=-3 or ah808=-3
then do;
  r5hrs80=-3;
end;
if sah801=-3 or sah802=-3 or sah803=-3 or sah804=-3 or sah805=-3 or sah806=-3 or sah807=-3 or
sah808=-3 then do;
  r5shrs80=-3;
end;
if eah801=-3 or eah802=-3 or eah803=-3 or eah804=-3 or eah805=-3 or eah806=-3 or eah807=-3 or
eah808=-3 then do;
  r5ehrs80=-3;
end;

```

/\* At this point the program repeats the above 1980 code for each year through 2002. This code is omitted here due to space considerations. Users who need to see the entire program should contact NLS User Services. The week numbers and variables created for each year are as follows:

Year/ Weeks	Variables Created		Year/ Weeks	Variables Created	
1981 53-104	wks811-wks818; swks811-swks818; ewks811-ewks818;	ah811-ah818; sah811-sah818; eah811-eah818;	1992 627-678	wks921-wks928; swks921-swks928; ewks921-ewks928;	ah921-ah928; sah921-sah928; eah921-eah928;
1982 105-156	wks821-wks828; swks821-swks828; ewks821-ewks828;	ah821-ah828; sah821-sah828; eah821-eah828;	1993 679-730	wks931-wks938; swks931-swks938; ewks931-ewks938;	ah931-ah938; sah931-sah938; eah931-eah938;
1983 157-209	wks831-wks838; swks831-swks838; ewks831-ewks838;	ah831-ah838; sah831-sah838; eah831-eah838;	1994 731-783	wks941-wks948; swks941-swks948; ewks941-ewks948;	ah941-ah948; sah941-sah948; eah941-eah948;
1984 210-261	wks841-wks848; swks841-swks848; ewks841-ewks848;	ah841-ah848; sah841-sah848; eah841-eah848;	1995 784-835	wks951-wks958; swks951-swks958; ewks951-ewks958;	ah951-ah958; sah951-sah958; eah951-eah958;
1985 262-313	wks851-wks858; swks851-swks858; ewks851-ewks858;	ah851-ah858; sah851-sah858; eah851-eah858;	1996 836-887	wks961-wks968; swks961-swks968; ewks961-ewks968;	ah961-ah968; sah961-sah968; eah961-eah968;
1986 314-365	wks861-wks868; swks861-swks868; ewks861-ewks868;	ah861-ah868; sah861-sah868; eah861-eah868;	1997 888-939	wks971-wks978; swks971-swks978; ewks971-ewks978;	ah971-ah978; sah971-sah978; eah971-eah978;
1987 366-417	wks871-wks878; swks871-swks878; ewks871-ewks878;	ah871-ah878; sah871-sah878; eah871-eah878;	1998 940-991	wks981-wks988; swks981-swks988; ewks981-ewks988;	ah981-ah988; sah981-sah988; eah981-eah988;
1988 418-470	wks881-wks888; swks881-swks888; ewks881-ewks888;	ah881-ah888; sah881-sah888; eah881-eah888;	1999 992-1043	wks991-wks998; swks991-swks998; ewks991-ewks998;	ah991-ah998; sah991-sah998; eah991-eah998;
	wks891-wks898;	ah891-ah898;		wks001-wks008;	ah001-ah008;

1989 471-522	swks891-swks898; ewks891-ewks898;	sah891-sah898; eah891-eah898;	2000 1044-1096	swks001-swks008; ewks001-ewks008;	sah001-sah008; eah001-eah008;
1990 523-574	wks901-wks908; swks901-swks908; ewks901-ewks908;	ah901-ah908; sah901-sah908; eah901-eah908;	2001 1097-1148	wks011-wks018; swks011-swks018; ewks011-ewks018;	ah011-ah018; sah011-sah018; eah011-eah018;
1991 575-626	wks911-wks918; swks911-swks918; ewks911-ewks918;	ah911-ah918; sah911-sah918; eah911-eah918;	2002 1149-1200	wks021-wks018; swks021-swks018; ewks021-ewks018;	ah021-ah028; sah021-sah028; eah021-eah028;

if e200 in (-3,-5) then do;

```
r5hrs80=-5; r5hrs81=-5; r5hrs82=-5; r5hrs83=-5; r5hrs84=-5; r5hrs85=-5;
r5hrs86=-5; r5hrs87=-5; r5hrs88=-5; r5hrs89=-5; r5hrs90=-5; r5hrs91=-5;
r5hrs92=-5; r5hrs93=-5; r5hrs94=-5; r5hrs95=-5; r5hrs96=-5; r5hrs97=-5;
r5hrs98=-5; r5hrs99=-5; r5hrs00=-5; r5hrs01=-5; r5hrs02=-5;
r5shrs80=-5; r5shrs81=-5; r5shrs82=-5; r5shrs83=-5; r5shrs84=-5; r5shrs85=-5;
r5shrs86=-5; r5shrs87=-5; r5shrs88=-5; r5shrs89=-5; r5shrs90=-5; r5shrs91=-5;
r5shrs92=-5; r5shrs93=-5; r5shrs94=-5; r5shrs95=-5; r5shrs96=-5; r5shrs97=-5;
r5shrs98=-5; r5shrs99=-5; r5shrs00=-5; r5shrs01=-5; r5shrs02=-5;
r5ehrs80=-5; r5ehrs81=-5; r5ehrs82=-5; r5ehrs83=-5; r5ehrs84=-5; r5ehrs85=-5;
r5ehrs86=-5; r5ehrs87=-5; r5ehrs88=-5; r5ehrs89=-5; r5ehrs90=-5; r5ehrs91=-5;
r5ehrs92=-5; r5ehrs93=-5; r5ehrs94=-5; r5ehrs95=-5; r5ehrs96=-5; r5ehrs97=-5;
r5ehrs98=-5; r5ehrs99=-5; r5ehrs00=-5; r5ehrs01=-5; r5ehrs02=-5;
```

end;

/\* Fill in -3's for cases where start/stop date is unknown/refused. \*/

do k=1 to 8;

/\* start date invalid \*/

```
if starfl=1 and r5hrs80>0 and stopw>1 then do; r5hrs80=-3; end;
if starfl=1 and r5hrs81>0 and stopw>52 then do; r5hrs81=-3; end;
if starfl=1 and r5hrs82>0 and stopw>104 then do; r5hrs82=-3; end;
if starfl=1 and r5hrs83>0 and stopw>156 then do; r5hrs83=-3; end;
if starfl=1 and r5hrs84>0 and stopw>209 then do; r5hrs84=-3; end;
if starfl=1 and r5hrs85>0 and stopw>261 then do; r5hrs85=-3; end;
if starfl=1 and r5hrs86>0 and stopw>313 then do; r5hrs86=-3; end;
if starfl=1 and r5hrs87>0 and stopw>365 then do; r5hrs87=-3; end;
if starfl=1 and r5hrs88>0 and stopw>417 then do; r5hrs88=-3; end;
if starfl=1 and r5hrs89>0 and stopw>470 then do; r5hrs89=-3; end;
if starfl=1 and r5hrs90>0 and stopw>522 then do; r5hrs90=-3; end;
if starfl=1 and r5hrs91>0 and stopw>574 then do; r5hrs91=-3; end;
if starfl=1 and r5hrs92>0 and stopw>626 then do; r5hrs92=-3; end;
if starfl=1 and r5hrs93>0 and stopw>678 then do; r5hrs93=-3; end;
if starfl=1 and r5hrs94>0 and stopw>730 then do; r5hrs94=-3; end;
if starfl=1 and r5hrs95>0 and stopw>783 then do; r5hrs95=-3; end;
if starfl=1 and r5hrs96>0 and stopw>835 then do; r5hrs96=-3; end;
if starfl=1 and r5hrs97>0 and stopw>887 then do; r5hrs97=-3; end;
if starfl=1 and r5hrs98>0 and stopw>939 then do; r5hrs98=-3; end;
if starfl=1 and r5hrs99>0 and stopw>990 then do; r5hrs99=-3; end;
if starfl=1 and r5hrs00>0 and stopw>1043 then do; r5hrs00=-3; end;
```

```

if starfl=1 and r5hrs01>0 and stopw>1096 then do; r5hrs01=-3; end;
if starfl=1 and r5hrs02>0 and stopw>1148 then do; r5hrs02=-3; end;

/* stop date invalid */
if stopfl=1 and r5hrs80>0 and starw<53 then do; r5hrs80=-3; end;
if stopfl=1 and r5hrs81>0 and starw<105 then do; r5hrs81=-3; end;
if stopfl=1 and r5hrs82>0 and starw<157 then do; r5hrs82=-3; end;
if stopfl=1 and r5hrs83>0 and starw<210 then do; r5hrs83=-3; end;
if stopfl=1 and r5hrs84>0 and starw<262 then do; r5hrs84=-3; end;
if stopfl=1 and r5hrs85>0 and starw<314 then do; r5hrs85=-3; end;
if stopfl=1 and r5hrs86>0 and starw<366 then do; r5hrs86=-3; end;
if stopfl=1 and r5hrs87>0 and starw<418 then do; r5hrs87=-3; end;
if stopfl=1 and r5hrs88>0 and starw<471 then do; r5hrs88=-3; end;
if stopfl=1 and r5hrs89>0 and starw<523 then do; r5hrs89=-3; end;
if stopfl=1 and r5hrs90>0 and starw<575 then do; r5hrs90=-3; end;
if stopfl=1 and r5hrs91>0 and starw<627 then do; r5hrs91=-3; end;
if stopfl=1 and r5hrs92>0 and starw<679 then do; r5hrs92=-3; end;
if stopfl=1 and r5hrs93>0 and starw<731 then do; r5hrs93=-3; end;
if stopfl=1 and r5hrs94>0 and starw<784 then do; r5hrs94=-3; end;
if stopfl=1 and r5hrs95>0 and starw<836 then do; r5hrs95=-3; end;
if stopfl=1 and r5hrs96>0 and starw<888 then do; r5hrs96=-3; end;
if stopfl=1 and r5hrs97>0 and starw<940 then do; r5hrs97=-3; end;
if stopfl=1 and r5hrs98>0 and starw<991 then do; r5hrs98=-3; end;
if stopfl=1 and r5hrs99>0 and starw<1044 then do; r5hrs99=-3; end;
if stopfl=1 and r5hrs00>0 and starw<1096 then do; r5hrs00=-3; end;
if stopfl=1 and r5hrs01>0 and starw<1149 then do; r5hrs01=-3; end;
if stopfl=1 and r5hrs02>0 and starw<1201 then do; r5hrs02=-3; end;
end;

```

/\*Merge Round 5 data with previous rounds \*/

```

array r4hrs r4hrs80-r4hrs99 r4hrs00 r4hrs01; /* Round 4 variable */
array thrs thrs80-thrs99 thrs00 thrs01; /* Total created variable */
array r4ehrs r4ehrs80-r4ehrs99 r4ehrs00 r4ehrs01;
array tehrs tehrs80-tehrs99 tehrs00 tehrs01;
array r4shrs r4shrs80-r4shrs99 r4shrs00 r4shrs01;
array tshrs tshrs80-tshrs99 tshrs00 tshrs01;

```

do over thrs; r4hrs=thrs; r4ehrs=tehrs; r4shrs=tshrs; end;

```

array thrs (i) thrs80-thrs99 thrs00-thrs02;
array tshrs (i) tshrs80-tshrs99 tshrs00-tshrs02;
array tehrs (i) tehrs80-tehrs99 tehrs00-tehrs02;
array r1hrs (i) r1hrs80-r1hrs99 r1hrs00-r1hrs02;
array r2hrs (i) r2hrs80-r2hrs99 r2hrs00-r2hrs02;
array r3hrs (i) r3hrs80-r3hrs99 r3hrs00-r3hrs02;
array r4hrs (i) r4hrs80-r4hrs99 r4hrs00-r4hrs02;
array r4shrs (i) r4shrs80-r4shrs99 r4shrs00-r4shrs02;
array r4ehrs (i) r4ehrs80-r4ehrs99 r4ehrs00-r4ehrs02;
array r5hrs (i) r5hrs80-r5hrs99 r5hrs00-r5hrs02;
array r5shrs (i) r5shrs80-r5shrs99 r5shrs00-r5shrs02;
array r5ehrs (i) r5ehrs80-r5ehrs99 r5ehrs00-r5ehrs02;

```

```
/* Begin by splitting up periods where Round 5 and dli Round exclusively collect hours worked  
information. Any hours worked information collected before the dli Round interview year should be  
independent of information collected in Round 5. */
```

```
/* For the new Round 5 c.v.'s, we are assuming that all dli jobs are employee-type jobs. Thus, dli  
information will be added to the "all jobs" and "employee jobs" c.v.'s but not "self-employed jobs" c.v.'s.  
*/
```

```
do over r5hrs;  
    thrs=0;  
    tshrs=0;  
    tehrs=0;  
    if r5hrs>0 then do; thrs=r5hrs; end;  
    if r5shrs>0 then do; tshrs=r5shrs; end;  
    if r5ehrs>0 then do; tehrs=r5ehrs; end;  
    if r4hrs>0 and dliwk=r4int then do; thrs=r4hrs; end;  
    if r4shrs>0 and dliwk=r4int then do; tshrs=r4shrs; end;  
    if r4ehrs>0 and dliwk=r4int then do; tehrs=r4ehrs; end;  
    if r3hrs>0 and dliwk=r3int then do; thrs=r3hrs; tehrs=r3hrs; end;  
    if r2hrs>0 and dliwk=r2int then do; thrs=r2hrs; tehrs=r2hrs; end;  
    if r1hrs>0 and dliwk=r1int then do; thrs=r1hrs; tehrs=r1hrs; end;  
    if r4hrs>0 and r5hrs>0 and dliwk=r4int then do; thrs=r4hrs+r5hrs; end;  
    if r4shrs>0 and r5shrs>0 and dliwk=r4int then do; tshrs=r4shrs+r5shrs; end;  
    if r4ehrs>0 and r5ehrs>0 and dliwk=r4int then do; tehrs=r4ehrs+r5ehrs; end;  
    if r3hrs>0 and r5hrs>0 and dliwk=r3int then do; thrs=r3hrs+r5hrs; end;  
    if r3hrs>0 and r5ehrs>0 and dliwk=r3int then do; tehrs=r3hrs+r5ehrs; end;  
    if r2hrs>0 and r5hrs>0 and dliwk=r2int then do; thrs=r2hrs+r5hrs; end;  
    if r2hrs>0 and r5ehrs>0 and dliwk=r2int then do; tehrs=r2hrs+r5ehrs; end;  
    if r1hrs>0 and r5hrs>0 and dliwk=r1int then do; thrs=r1hrs+r5hrs; end;  
    if r1hrs>0 and r5ehrs>0 and dliwk=r1int then do; tehrs=r1hrs+r5ehrs; end;  
end;
```

```
do over r5hrs;  
    if -4<r5hrs<0 then do; thrs=r5hrs; end;  
    if -4<r5shrs<0 then do; tshrs=r5shrs; end;  
    if -4<r5ehrs<0 then do; tehrs=r5ehrs; end;  
    if -4<r4hrs<0 then do; thrs=r4hrs; end;  
    if -4<r4shrs<0 then do; tshrs=r4shrs; end;  
    if -4<r4ehrs<0 then do; tehrs=r4ehrs; end;  
    if -4<r3hrs<0 and dliwk=r3int then do; thrs=r3hrs; tehrs=r3hrs; end;  
    if -4<r2hrs<0 and dliwk=r2int then do; thrs=r2hrs; tehrs=r2hrs; end;  
    if -4<r1hrs<0 and dliwk=r1int then do; thrs=r1hrs; tehrs=r1hrs; end;  
end;
```

```
do over thrs;  
    if e200=-5 then do; thrs=-5; tehrs=-5; tshrs=-5; end;  
end;
```

```
endsas;
```

---

## Total Hours Worked Age 14-19, Hours Worked at All Jobs Since Age 20, and Hours Worked at Employee-Type Jobs Since Age 20

**Variables Created:** CV\_HOURS\_WK\_TEEN  
CV\_HOURS\_WK\_ADULT\_ALL  
CV\_HOURS\_WK\_ADULT\_ET

### Programs Used

This program uses [emp\\_begin.sas](#) as input (see the first page of this appendix for details).

This program calculates the number of hours worked by the respondent at all employee-type jobs between the ages of 14 and 19. It then creates two variables for the number of hours worked by the respondent since age 20: one for all jobs and one for employee-type jobs. **See the [introduction to this appendix for important information about the universes for each of these variables](#).** A variable is created for each respondent even if the respondent has worked no jobs in a given year with the default value set to zero (0). Note that when both "starting hours" and "current hours" are reported, the latter are used to construct these measures.

---

```
*** Organize hours for each job;
array starw (i) starw1-starw8;
array stopw (i) stopw1-stopw8;
array starfl (i) starfl_1-starfl_8;
array stopfl (i) stopfl_1-stopfl_8;

/* shrs1 - starting hours
   ehrs1 - ending hours */

array shrs (i) shrs1-shrs8;
array ehrs (i) ehrs1-ehrs8;
array hours (i) hours1-hours8;
array hrck (i) hrck1-hrck8;

array e22610 (i) e226101-e226108;                                array e22611 (i) e226111-e226118;
array e100231 (i) e1002311-e1002318;                            array e100232 (i) e1002321-e1002328;
array e23901 (i) e239011-e239018;                               array e24501 (i) e245011-e245018;
array e34402 (i) e344021-e344028;                               array e34403 (i) e344031-e344038;
array e34428 (i) e344281-e344288;                             array e37904 (i) e379041-e379048;
array e38000f (i) e38000f1-e38000f8;                           array e38103 (i) e381031-e381038;
array e38105 (i) e381051-e381058;                             array e59901 (i) e599011-e599018;
array e88000 (i) e880001-e880008;                            array e88501 (i) e885011-e885018;
array e98402 (i) e984021-e984208;                           array e98403 (i) e984031-e984038;
array e98429 (i) e984291-e984298;                          array e100256 (i) e1002561-e1002568;

/* Define the number of hours per week worked at the start date and the end date of a job. */
do i=1 to 8;
  if e23901>-4 then do; shrs=e23901; end;
```

```

if e23901>-4 and e24501>-4 then do; shrs=e23901+e24501; end;
if e34402>-4 then do; shrs=e34402; end;
if e34403>-4 then do; shrs=e34403; end;
if e34403>-4 and e34428>-4 then do; shrs=e34403+e34428; end;
if e37904>-4 then do; ehrs=e37904; end;
if e38000f>-4 then do; ehrs=e38000f; end;
if e38103>-4 then do; ehrs=e38103; end;
if e38103>-4 and e38105>-4 then do; ehrs=e38103+e38105; end;
if e59901>-4 then do; shrs=e59901; end;
if e88000>-4 then do; shrs=e88000; end;
if e88501>-4 and e88000>-4 then do; shrs=e88501+e88000; end;
if e98402>-4 then do; shrs=e98402; ed;
if e98429>-4 then do; shrs=e98429; end;
if e98429>-4 and e98403>-4 then do; shrs=e98403+e98429; end;
if e22610>-4 then do; shrs=e22610; end;
if e22611>-4 then do; shrs=e22611; end;
if e22610>-4 and e22611>-4 then do; shrs=e22610+e22611; end;
if e100231>-4 then do; shrs=e100231; end;
if e100231>-4 then do; shrs=e100232; end;
if e100231>-4 and e100232>-4 then do; shrs=e100231+e100232; end;
if e100256>-4 then do; shrs=e100256; end;
end;

```

/\* The code below decides which hours per week total (starting or ending) will be used in determining hours worked per year. Hours per week starts with a default number (0), is updated if starting hours are available, and is updated once more if ending hours are available. \*/

\* Set default hours to zero (-16);

do over hours;

  hours=0;

  hrck=0;

end;

\* Take starting hours if reported (eliminates -4's);

do over shrs;

  if shrs=>0 then do; hours=shrs; end;

  if -4<shrs<0 then do; hrck=shrs; end;

end;

\* Write over if end hours reported;

do over ehrs;

  if ehrs=>0 then do; hours=ehrs; end;

  if -4<ehrs<0 then do; hrck=ehrs; end;

end;

array job1wks (i) wk1\_1-wk1\_1200;

array job3wks (i) wk3\_1-wk3\_1200;

array job5wks (i) wk5\_1-wk5\_1200;

array job7wks (i) wk7\_1-wk7\_1200;

array job2wks (i) wk2\_1-wk2\_1200;

array job4wks (i) wk4\_1-wk4\_1200;

array job6wks (i) wk6\_1-wk6\_1200;

array job8wks (i) wk8\_1-wk8\_1200;

\*\* Calculate cumulative weeks on employee-type jobs for each year \*\*;

```

/* Initialize created variables */
r5hrs14=0;
r5hrs20=0;
r5ehrs20=0;

/* Read "wks145" as the number of weeks worked between the ages of 14 and 19, for job 5 only. Read
"thrs145" as the number of hours worked between the ages of 14 and 19 for job 5 only. */
wks141=0; wks142=0; wks143=0; wks144=0; wks145=0; wks146=0; wks147=0; wks148=0; thrs141=0;
thrs142=0; thrs143=0; thrs144=0; thrs145=0; thrs146=0; thrs147=0; thrs148=0;

/* Respondents up by those who have not turned 20 by Round 4. */
if age14wk>0 and (age20wk-1)>1200 then do;
do i=age14wk to 1200;
  if job1wks=1 and self1=0 then do; wks141=wks141+1; end;
  if job2wks=1 and self2=0 then do; wks142=wks142+1; end;
  if job3wks=1 and self3=0 then do; wks143=wks143+1; end;
  if job4wks=1 and self4=0 then do; wks144=wks144+1; end;
  if job5wks=1 and self5=0 then do; wks145=wks145+1; end;
  if job6wks=1 and self6=0 then do; wks146=wks146+1; end;
  if job7wks=1 and self7=0 then do; wks147=wks147+1; end;
  if job8wks=1 and self8=0 then do; wks148=wks148+1; end;
end;

do i=age14wk to 1200;
  if job1wks ne -3 and hours1=>0 and self1=0 then thrs141=hours1*wks141;
  if job2wks ne -3 and hours2=>0 and self2=0 then thrs142=hours2*wks142;
  if job3wks ne -3 and hours3=>0 and self3=0 then thrs143=hours3*wks143;
  if job4wks ne -3 and hours4=>0 and self4=0 then thrs144=hours4*wks144;
  if job5wks ne -3 and hours5=>0 and self5=0 then thrs145=hours5*wks145;
  if job6wks ne -3 and hours6=>0 and self6=0 then thrs146=hours6*wks146;
  if job7wks ne -3 and hours7=>0 and self7=0 then thrs147=hours7*wks147;
  if job8wks ne -3 and hours8=>0 and self8=0 then thrs148=hours8*wks148;
end;

do i=age14wk to 1200;
  if (job1wks=-3 and self1=0) then thrs141=-3;
  if (job2wks=-3 and self2=0) then thrs142=-3;
  if (job3wks=-3 and self3=0) then thrs143=-3;
  if (job4wks=-3 and self4=0) then thrs144=-3;
  if (job5wks=-3 and self5=0) then thrs145=-3;
  if (job6wks=-3 and self6=0) then thrs146=-3;
  if (job7wks=-3 and self7=0) then thrs147=-3;
  if (job8wks=-3 and self8=0) then thrs148=-3;
end;

if thrs141=>0 and thrs142=>0 and thrs143=>0 and thrs144=>0 and thrs145=>0 and thrs146=>0 and
thrs147=>0 and thrs148=>0 then do;
  r5hrs14=thrs141+thrs142+thrs143+thrs144+thrs145+thrs146+thrs147+thrs148;
end;

```

```

if (thrs141=-3 and self1=0) or (thrs142=-3 and self2=0) or (thrs143=-3 and self3=0) or (thrs144=-3 and
self4=0)
or (thrs145=-3 and self5=0) or (thrs146=-3 and self6=0) or (thrs147=-3 and self7=0) or (thrs148=-3
and self8=0)
then do;
  r5hrs14=-3;
end;

```

```

/* hrck is used when invalid hrs/wk answers are given */
if wks141>0 and hrck1<0 and r5hrs14>0 and self1=0 then r5hrs14=-3;
if wks142>0 and hrck2<0 and r5hrs14>0 and self2=0 then r5hrs14=-3;
if wks143>0 and hrck3<0 and r5hrs14>0 and self3=0 then r5hrs14=-3;
if wks144>0 and hrck4<0 and r5hrs14>0 and self4=0 then r5hrs14=-3;
if wks145>0 and hrck5<0 and r5hrs14>0 and self5=0 then r5hrs14=-3;
if wks146>0 and hrck6<0 and r5hrs14>0 and self6=0 then r5hrs14=-3;
if wks147>0 and hrck7<0 and r5hrs14>0 and self7=0 then r5hrs14=-3;
if wks148>0 and hrck8<0 and r5hrs14>0 and self8=0 then r5hrs14=-3;

```

```
end;
```

```
/* Respondents up by those who have turned 20 by Round 3. */
```

```

if age14wk>0 and (age20wk-1)<=1200 then do;
do i=age14wk to (age20wk-1);
  if job1wks=1 and self1=0 then do; wks141=wks141+1; end;
  if job2wks=1 and self2=0 then do; wks142=wks142+1; end;
  if job3wks=1 and self3=0 then do; wks143=wks143+1; end;
  if job4wks=1 and self4=0 then do; wks144=wks144+1; end;
  if job5wks=1 and self5=0 then do; wks145=wks145+1; end;
  if job6wks=1 and self6=0 then do; wks146=wks146+1; end;
  if job7wks=1 and self7=0 then do; wks147=wks147+1; end;
  if job8wks=1 and self8=0 then do; wks148=wks148+1; end;
end;

```

```
do i=age14wk to (age20wk-1);
```

```

  if job1wks ne -3 and hours1=>0 and self1=0 then thrs141=hours1*wks141;
  if job2wks ne -3 and hours2=>0 and self2=0 then thrs142=hours2*wks142;
  if job3wks ne -3 and hours3=>0 and self3=0 then thrs143=hours3*wks143;
  if job4wks ne -3 and hours4=>0 and self4=0 then thrs144=hours4*wks144;
  if job5wks ne -3 and hours5=>0 and self5=0 then thrs145=hours5*wks145;
  if job6wks ne -3 and hours6=>0 and self6=0 then thrs146=hours6*wks146;
  if job7wks ne -3 and hours7=>0 and self7=0 then thrs147=hours7*wks147;
  if job8wks ne -3 and hours8=>0 and self8=0 then thrs148=hours8*wks148;
end;

```

```
do i=age14wk to (age20wk-1);
```

```

  if (job1wks=-3 and self1=0) then thrs141=-3;
  if (job2wks=-3 and self2=0) then thrs142=-3;
  if (job3wks=-3 and self3=0) then thrs143=-3;
  if (job4wks=-3 and self4=0) then thrs144=-3;
  if (job5wks=-3 and self5=0) then thrs145=-3;
  if (job6wks=-3 and self6=0) then thrs146=-3;

```

```
if (job7wks=-3 and self7=0) then thrs147=-3;
if (job8wks=-3 and self8=0) then thrs148=-3;
end;
```

```
if thrs141=>0 and thrs142=>0 and thrs143=>0 and thrs144=>0 and thrs145=>0 and thrs146=>0 and
thrs147=>0 and
thrs148=>0 then do;
r5hrs14=thrs141+thrs142+thrs143+thrs144+thrs145+thrs146+thrs147+thrs148;
end;
```

```
if (thrs141=-3 and self1=0) or (thrs142=-3 and self2=0) or (thrs143=-3 and self3=0) or (thrs144=-3 and
self4=0)
or (thrs145=-3 and self5=0) or (thrs146=-3 and self6=0) or (thrs147=-3 and self7=0) or (thrs148=-3
and self8=0)
then do;
r5hrs14=-3;
end;
```

```
/* hrck is used when invalid hrs./wk answers are given */
if wks141>0 and hrck1<0 and r5hrs14>0 and self1=0 then r5hrs14=-3;
if wks142>0 and hrck2<0 and r5hrs14>0 and self2=0 then r5hrs14=-3;
if wks143>0 and hrck3<0 and r5hrs14>0 and self3=0 then r5hrs14=-3;
if wks144>0 and hrck4<0 and r5hrs14>0 and self4=0 then r5hrs14=-3;
if wks145>0 and hrck5<0 and r5hrs14>0 and self5=0 then r5hrs14=-3;
if wks146>0 and hrck6<0 and r5hrs14>0 and self6=0 then r5hrs14=-3;
if wks147>0 and hrck7<0 and r5hrs14>0 and self7=0 then r5hrs14=-3;
if wks148>0 and hrck8<0 and r5hrs14>0 and self8=0 then r5hrs14=-3;

end;
```

```
/* Read "wks205" as the number of weeks worked after age 20 for job 5 only. Read "thrs205" as the
number of hours worked after age 20 for job 5 only. */
wks201=0; wks202=0; wks203=0; wks204=0; wks205=0; wks206=0; wks207=0; wks208=0;
thrs201=0; thrs202=0; thrs203=0; thrs204=0; thrs205=0; thrs206=0; thrs207=0; thrs208=0;
```

```
/* Respondents up by those who have turned 20 by Round 3. */
if age14wk>0 and (age20wk-1)<=1200 then do;
do i=age20wk to 1200;
if job1wks=1 then do; wks201=wks201+1; end;
if job2wks=1 then do; wks202=wks202+1; end;
if job3wks=1 then do; wks203=wks203+1; end;
if job4wks=1 then do; wks204=wks204+1; end;
if job5wks=1 then do; wks205=wks205+1; end;
if job6wks=1 then do; wks206=wks206+1; end;
if job7wks=1 then do; wks207=wks207+1; end;
if job8wks=1 then do; wks208=wks208+1; end;
end;
```

```
do i=age20wk to 1200;
if job1wks ne -3 and hours1=>0 then thrs201=hours1*wks201;
if job2wks ne -3 and hours2=>0 then thrs202=hours2*wks202;
```

```

if job3wks ne -3 and hours3=>0 then thrs203=hours3*wks203;
if job4wks ne -3 and hours4=>0 then thrs204=hours4*wks204;
if job5wks ne -3 and hours5=>0 then thrs205=hours5*wks205;
if job6wks ne -3 and hours6=>0 then thrs206=hours6*wks206;
if job7wks ne -3 and hours7=>0 then thrs207=hours7*wks207;
if job8wks ne -3 and hours8=>0 then thrs208=hours8*wks208;
end;

do i=age20wk to 1200;
  if (job1wks=-3) then thrs201=-3;
  if (job2wks=-3) then thrs202=-3;
  if (job3wks=-3) then thrs203=-3;
  if (job4wks=-3) then thrs204=-3;
  if (job5wks=-3) then thrs205=-3;
  if (job6wks=-3) then thrs206=-3;
  if (job7wks=-3) then thrs207=-3;
  if (job8wks=-3) then thrs208=-3;
end;

if thrs201=>0 and thrs202=>0 and thrs203=>0 and thrs204=>0 and thrs205=>0 and thrs206=>0 and
thrs207=>0 and
thrs208=>0 then do;
  r5hrs20=thrs201+thrs202+thrs203+thrs204+thrs205+thrs206+thrs207+thrs208;
end;

if thrs201=-3 or thrs202=-3 or thrs203=-3 or thrs204=-3 or thrs205=-3 or thrs206=-3 or thrs207=-3 or
thrs208=-3
then do;
  r5hrs20=-3;
end;

/* hrck is used when invalid hrs./wk answers are given */
if wks201>0 and hrck1<0 and r5hrs20>0 then r5hrs20=-3;
if wks202>0 and hrck2<0 and r5hrs20>0 then r5hrs20=-3;
if wks203>0 and hrck3<0 and r5hrs20>0 then r5hrs20=-3;
if wks204>0 and hrck4<0 and r5hrs20>0 then r5hrs20=-3;
if wks205>0 and hrck5<0 and r5hrs20>0 then r5hrs20=-3;
if wks206>0 and hrck6<0 and r5hrs20>0 then r5hrs20=-3;
if wks207>0 and hrck7<0 and r5hrs20>0 then r5hrs20=-3;
if wks208>0 and hrck8<0 and r5hrs20>0 then r5hrs20=-3;

end;

/* Read "wks205" as the number of weeks worked after age 20 for job 5 only. Read "thrs205" as the
number of hours worked after age 20 for job 5 only. */
ewks201=0; ewks202=0; ewks203=0; ewks204=0; ewks205=0; ewks206=0; ewks207=0; ewks208=0;
tehrs201=0; tehrs202=0; tehrs203=0; tehrs204=0; tehrs205=0; tehrs206=0; tehrs207=0; tehrs208=0;

/* Respondents up by those who have turned 20 by Round 3. */
if age14wk>0 and (age20wk-1)<=1200 then do;
  do i=age20wk to 1200;

```

```
if job1wks=1 and self1=0 then do; ewks201=ewks201+1; end;
if job2wks=1 and self2=0 then do; ewks202=ewks202+1; end;
if job3wks=1 and self3=0 then do; ewks203=ewks203+1; end;
if job4wks=1 and self4=0 then do; ewks204=ewks204+1; end;
if job5wks=1 and self5=0 then do; ewks205=ewks205+1; end;
if job6wks=1 and self6=0 then do; ewks206=ewks206+1; end;
if job7wks=1 and self7=0 then do; ewks207=ewks207+1; end;
if job8wks=1 and self8=0 then do; ewks208=ewks208+1; end;
end;
```

```
do i=age20wk to 1200;
  if job1wks ne -3 and hours1=>0 and self1=0 then tehrs201=hours1*ewks201;
  if job2wks ne -3 and hours2=>0 and self2=0 then tehrs202=hours2*ewks202;
  if job3wks ne -3 and hours3=>0 and self3=0 then tehrs203=hours3*ewks203;
  if job4wks ne -3 and hours4=>0 and self4=0 then tehrs204=hours4*ewks204;
  if job5wks ne -3 and hours5=>0 and self5=0 then tehrs205=hours5*ewks205;
  if job6wks ne -3 and hours6=>0 and self6=0 then tehrs206=hours6*ewks206;
  if job7wks ne -3 and hours7=>0 and self7=0 then tehrs207=hours7*ewks207;
  if job8wks ne -3 and hours8=>0 and self8=0 then tehrs208=hours8*ewks208;
end;
```

```
do i=age20wk to 1200;
  if (job1wks=-3 and self1=0) then tehrs201=-3;
  if (job2wks=-3 and self2=0) then tehrs202=-3;
  if (job3wks=-3 and self3=0) then tehrs203=-3;
  if (job4wks=-3 and self4=0) then tehrs204=-3;
  if (job5wks=-3 and self5=0) then tehrs205=-3;
  if (job6wks=-3 and self6=0) then tehrs206=-3;
  if (job7wks=-3 and self7=0) then tehrs207=-3;
  if (job8wks=-3 and self8=0) then tehrs208=-3;
end;
```

```
if tehrs201=>0 and tehrs202=>0 and tehrs203=>0 and tehrs204=>0 and tehrs205=>0 and tehrs206=>0
and tehrs207=>0
and tehrs208=>0 then do;
  r5ehrs20=tehrs201+tehrs202+tehrs203+tehrs204+tehrs205+tehrs206+tehrs207+tehrs208;
end;
```

```
if tehrs201=-3 or tehrs202=-3 or tehrs203=-3 or tehrs204=-3 or tehrs205=-3 or tehrs206=-3 or
tehrs207=-3
or tehrs208=-3 then do;
  r5ehrs20=-3;
end;
```

```
/* hrck is used when invalid hrs./wk answers are given */
if ewks201>0 and hrck1<0 and r5ehrs20>0 and self1=0 then r5ehrs20=-3;
if ewks202>0 and hrck2<0 and r5ehrs20>0 and self2=0 then r5ehrs20=-3;
if ewks203>0 and hrck3<0 and r5ehrs20>0 and self3=0 then r5ehrs20=-3;
if ewks204>0 and hrck4<0 and r5ehrs20>0 and self4=0 then r5ehrs20=-3;
if ewks205>0 and hrck5<0 and r5ehrs20>0 and self5=0 then r5ehrs20=-3;
if ewks206>0 and hrck6<0 and r5ehrs20>0 and self6=0 then r5ehrs20=-3;
```

```

if ewks207>0 and hrck7<0 and r5ehrs20>0 and self7=0 then r5ehrs20=-3;
if ewks208>0 and hrck8<0 and r5ehrs20>0 and self8=0 then r5ehrs20=-3;

end;
array self (i) self1-self8;

/* Remove invalid start/stop dates */
do i=1 to 8;
if age14wk<784 then do;
  if starfl=1 and r5hrs14>0 and stopw>730 then do; r5hrs14=-3; end;
end;
if age14wk<836 then do;
  if starfl=1 and r5hrs14>0 and stopw>783 then do; r5hrs14=-3; end;
end;
if age14wk<888 then do;
  if starfl=1 and r5hrs14>0 and stopw>835 then do; r5hrs14=-3; end;
end;
if age14wk<940 then do;
  if starfl=1 and r5hrs14>0 and stopw>887 then do; r5hrs14=-3; end;
end;
if age14wk<991 then do;
  if starfl=1 and r5hrs14>0 and stopw>939 then do; r5hrs14=-3; end;
end;
if age14wk<1044 then do;
  if starfl=1 and r5hrs14>0 and stopw>990 then do; r5hrs14=-3; end;
end;
if age14wk<1097 then do;
  if starfl=1 and r5hrs14>0 and stopw>1043 then do; r5hrs14=-3; end;
end;
if age20wk<1097 then do;
  if starfl=1 and r5hrs20>0 and stopw>1043 then do; r5hrs20=-3; end;
end;
if age20wk<1097 then do;
  if starfl=1 and self=0 and r5ehrs20>0 and stopw>1043 then do; r5ehrs20=-3; end;
end;
if age14wk<1149 then do;
  if starfl=1 and r5hrs14>0 and stopw>1096 then do; r5hrs14=-3; end;
end;
if age20wk<1149 then do;
  if starfl=1 and r5hrs20>0 and stopw>1096 then do; r5hrs20=-3; end;
end;
if age20wk<1149 then do;
  if starfl=1 and self=0 and r5ehrs20>0 and stopw>1096 then do; r5ehrs20=-3; end;
end;
if age14wk<1201 then do;
  if starfl=1 and r5hrs14>0 and stopw>1148 then do; r5hrs14=-3; end;
end;
if age20wk<1201 then do;
  if starfl=1 and r5hrs20>0 and stopw>1148 then do; r5hrs20=-3; end;
end;
if age20wk<1201 then do;

```

```

if starfl=1 and self=0 and r5ehrs20>0 and stopw>1148 then do; r5ehrs20=-3; end;
end;

if age14wk<784 then do;
  if stopfl=1 and r5hrs14>0 and starw<784 then do; r5hrs14=-3; end;
end;
if age14wk<836 then do;
  if stopfl=1 and r5hrs14>0 and starw<836 then do; r5hrs14=-3; end;
end;
if age14wk<888 then do;
  if stopfl=1 and r5hrs14>0 and starw<888 then do; r5hrs14=-3; end;
end;
if age14wk<940 then do;
  if stopfl=1 and r5hrs14>0 and starw<940 then do; r5hrs14=-3; end;
end;
if age14wk<991 then do;
  if stopfl=1 and r5hrs14>0 and starw<991 then do; r5hrs14=-3; end;
end;
if age14wk<1044 then do;
  if stopfl=1 and r5hrs14>0 and starw<1044 then do; r5hrs14=-3; end;
end;
if age14wk<1097 then do;
  if stopfl=1 and r5hrs14>0 and starw<1097 then do; r5hrs14=-3; end;
end;
if age20wk<1097 then do;
  if stopfl=1 and r5hrs20>0 and starw<1097 then do; r5hrs20=-3; end;
end;
if age20wk<1097 then do;
  if stopfl=1 and self=0 and r5ehrs20>0 and starw<1097 then do; r5ehrs20=-3; end;
end;
if age14wk<1149 then do;
  if stopfl=1 and r5hrs14>0 and starw<1149 then do; r5hrs14=-3; end;
end;
if age20wk<1149 then do;
  if stopfl=1 and r5hrs20>0 and starw<1149 then do; r5hrs20=-3; end;
end;
if age20wk<1149 then do;
  if stopfl=1 and self=0 and r5ehrs20>0 and starw<1149 then do; r5ehrs20=-3; end;
end;
if age14wk<1201 then do;
  if stopfl=1 and r5hrs14>0 and starw<1201 then do; r5hrs14=-3; end;
end;
if age20wk<1201 then do;
  if stopfl=1 and r5hrs20>0 and starw<1201 then do; r5hrs20=-3; end;
end;
if age20wk<1201 then do;
  if stopfl=1 and self=0 and r5ehrs20>0 and starw<1201 then do; r5ehrs20=-3; end;
end;
end;

if e200=-5 then do;

```

```
r5hrs14=-5;  
r5hrs20=-5;  
r5ehrs20=-5;  
end;
```

#### **/\* Merge with previous rounds \*/**

```
r4hrs14=thrs14;  
r4hrs20=thrs20;  
r4ehrs20=tehrs20;
```

```
/* Initialize created variables to zero. */
```

```
thrs14=0;  
thrs20=0;  
tehrs20=0;
```

```
/* By the construction of the dli created variables, we can simply add the two created variables from the current and previous rounds if they are both positive. If one is positive and one is not, then the positive value will be the total created variable for both rounds. If neither is positive, then the total created variable for both rounds will be zero. */
```

```
if r5hrs14=>0 then do; thrs14=r5hrs14; end;  
if r5hrs14=-3 then do; thrs14=-3; end;
```

```
/* dli is Round 4 */
```

```
if r4hrs14=>0 and r5hrs14=>0 then do; thrs14=r4hrs14+r5hrs14; end;  
if r4hrs14=-3 then do; thrs14=-3; end;
```

```
/* dli is Round 3 */
```

```
if r3hrs14=>0 and r5hrs14=>0 and r4int=-5 then do; thrs14=r3hrs14+r5hrs14; end;  
if r3hrs14=-3 and r4int=-5 then do; thrs14=-3; end;
```

```
/* dli is Round 2 */
```

```
if r2hrs14=>0 and r5hrs14=>0 and r3int=-5 and r4int=-5 then do; thrs14=r2hrs14+r5hrs14; end;  
if r2hrs14=-3 and r3int=-5 and r4int=-5 then do; thrs14=-3; end;
```

```
/* dli is Round 1 */
```

```
if r1hrs14=>0 and r4hrs14=>0 and r4int=-5 and r3int=-5 and r2int=-5 then do; thrs14=r1hrs14+r4hrs14;  
end;  
if r1hrs14=-3 and r3int=-5 and r2int=-5 and r4int=-5 then do; thrs14=-3; end;
```

```
if r5hrs20>0 then thrs20=r5hrs20;
```

```
if r4hrs20>0 then thrs20=r4hrs20;  
if r5hrs20>0 and r4hrs20>0 then do; thrs20=r5hrs20+r4hrs20; end;  
if r5hrs20=-3 or r4hrs20=-3 then do; thrs20=-3; end;
```

```
if r3hrs20>0 and r4int=-5 then thrs20=r3hrs20;
```

```
if r5hrs20>0 and r3hrs20>0 and r4int=-5 then do; thrs20=r5hrs20+r3hrs20; end;  
if r5hrs20=-3 or r3hrs20=-3 and r4int=-5 then do; thrs20=-3; end;
```

```
if r5ehrs20>0 then tehrs20=r5ehrs20;
```

```

if r4ehrs20>0 then tehrs20=r4ehrs20;
if r5ehrs20>0 and r4hrs20>0 then do; tehrs20=r5ehrs20+r4hrs20; end;
if r5ehrs20=-3 or r4hrs20=-3 then do; tehrs20=-3; end;

if r3hrs20>0 and r4int=-5 then tehrs20=r3hrs20;
if r5ehrs20>0 and r3hrs20>0 and r4int=-5 then do; tehrs20=r5ehrs20+r3hrs20; end;
if (r5ehrs20=-3 or r3hrs20=-3) and r4int=-5 then do; tehrs20=-3; end;

/* For respondents younger than 20 at interview date, their thrs14 c.v. will be -4. */
if age20wk>intwk and intwk>0 then do; thrs20=-4; tehrs20=-4; end;

/* Account for non-interview cases */
if uid1=-5 then do; thrs14=-5; thrs20=-5; tehrs20=-5; end;

endsas;

```

---

## Number of Jobs Held during 19xx/20xx

**Variables Created:** CV\_TTL\_JOB\_ALL.80 - CV\_TTL\_JOB\_ALL.02  
                  CV\_TTL\_JOB\_ET.80 - CV\_TTL\_JOB\_ET.02  
                  CV\_TTL\_JOB\_SE.98 - CV\_TTL\_JOB\_SE.02

### Programs Used

This program uses [emp\\_begin.sas](#) as input (see the first page of this appendix for details).

This program calculates the number of employee-type jobs the respondent held during each calendar year. Three sets of variables are created: one for all jobs, one for employee-type jobs, and one for self-employed jobs. **See the [introduction to this appendix for important information about the universes for each of these variables](#).** These variables are created only for respondents who have worked at least one week in the applicable calendar year.

---

```

array job1wks (i) wk1_1-wk1_1200;
array job2wks (i) wk2_1-wk2_1200;
array job3wks (i) wk3_1-wk3_1200;
array job4wks (i) wk4_1-wk4_1200;
array job5wks (i) wk5_1-wk5_1200;
array job6wks (i) wk6_1-wk6_1200;
array job7wks (i) wk7_1-wk7_1200;
array job8wks (i) wk8_1-wk8_1200;
array starw (i) starw1-starw8;
array stopw (i) stopw1-stopw8;
array starfl (i) starfl_1-starfl_8;
array stopfl (i) stopfl_1-stopfl_8;

```

\*\* Indicate if worked at least one week on a job in a given year \*\*;

```

/* 1980 */
r5j801=0;    r5s801=0;    r5e801=0;
r5j802=0;    r5s802=0;    r5e802=0;
r5j803=0;    r5s803=0;    r5e803=0;
r5j804=0;    r5s804=0;    r5e804=0;
r5j805=0;    r5s805=0;    r5e805=0;
r5j806=0;    r5s806=0;    r5e806=0;
r5j807=0;    r5s807=0;    r5e807=0;
r5j808=0;    r5s808=0;    r5e808=0;

do i=1 to 52;
  if job1wks=-3 then do; r5j801=-3; end;
  if job1wks=-3 and self1=1 then do; r5s801=-3; end;
  if job1wks=-3 and self1=0 then do; r5e801=-3; end;
  if job2wks=-3 then do; r5j802=-3; end;
  if job2wks=-3 and self2=1 then do; r5s802=-3; end;
  if job2wks=-3 and self2=0 then do; r5e802=-3; end;
  /* and so on through */
  if job8wks=-3 then do; r5j808=-3; end;
  if job8wks=-3 and self8=1 then do; r5s808=-3; end;
  if job8wks=-3 and self8=0 then do; r5e808=-3; end;
end;

do i=1 to 52;
  if job1wks=1 then do; r5j801=1; end;
  if job1wks=1 and self1=1 then do; r5s801=1; end;
  if job1wks=1 and self1=0 then do; r5e801=1; end;
  if job2wks=1 then do; r5j802=1; end;
  if job2wks=1 and self2=1 then do; r5s802=1; end;
  if job2wks=1 and self2=0 then do; r5e802=1; end;
  /* and so on through */
  if job8wks=1 then do; r5j808=1; end;
  if job8wks=1 and self8=1 then do; r5s808=1; end;
  if job8wks=1 and self8=0 then do; r5e808=1; end;
end;

if r5j801 ne -3 and r5j802 ne -3 and r5j803 ne -3 and r5j804 ne -3 and r5j805 ne -3 and r5j806 ne -3 and
r5j807 ne -3 and r5j808 ne -3 ne -3 then do;
  r5job80=sum (r5j801, r5j802, r5j803, r5j804, r5j805, r5j806, r5j807, r5j808);
end;
if r5j801=-3 or r5j802=-3 or r5j803=-3 or r5j804=-3 or r5j805=-3 or r5j806=-3 or r5j807=-3 or r5j808=-3 then do;
  r5job80=-3;
end;

if r5s801 ne -3 and r5s802 ne -3 and r5s803 ne -3 and r5s804 ne -3 and r5s805 ne -3 and r5s806 ne -3 and
r5s807 ne -3 and r5s808 ne -3 ne -3 then do;
  r5self80=sum (r5s801, r5s802, r5s803, r5s804, r5s805, r5s806, r5s807, r5s808);
end;
if r5s801=-3 or r5s802=-3 or r5s803=-3 or r5s804=-3 or r5s805=-3 or r5s806=-3 or r5s807=-3 or

```

```

r5s808=-3 then do;
  r5self80=-3;
end;

if r5e801 ne -3 and r5e802 ne -3 and r5e803 ne -3 and r5e804 ne -3 and r5e805 ne -3 and r5e806 ne -3
and r5e807 ne -3 and r5e808 ne -3 then do;
  r5emp80=sum (r5e801, r5e802, r5e803, r5e804, r5e805, r5e806, r5e807, r5e808);
end;
if r5e801=-3 or r5e802=-3 or r5e803=-3 or r5e804=-3 or r5e805=-3 or r5e806=-3 or r5e807=-3 or
r5e808=-3 then do;
  r5emp80=-3;
end;

```

/\* At this point the program repeats the above code for each year through 2002. The code is omitted here due to space considerations. Users should contact NLS User Services for more information. The week numbers for the "do i" statements are as follows:

1981	53-104	1992	627-678
1982	105-156	1993	679-730
1983	157-209	1994	731-783
1984	210-261	1995	784-835
1985	262-313	1996	836-887
1986	314-365	1997	888-939
1987	366-417	1998	940-991
1988	418-470	1999	992-1044
1989	471-522	2000	1045-1096
1990	523-574	2001	1097-1148
1991	575-626	2002	1149-1200***/

```

array self (i) self1-self8;

do i=1 to 8;
/* start date invalid */
if (stopfl=1 or starfl=1) then do;
  if self=1 and r5job80>0 then do;
    r5self80=-3; r5job80=-3; end;
  if self=0 and r5job80>0 then do;
    r5emp80=-3; r5job80=-3; end;
  if self=1 and r5job81>0 then do;
    r5self81=-3; r5job81=-3; end;
  if self=0 and r5job81>0 then do;
    r5emp81=-3; r5job81=-3; end;
    /*and so on through */
  if self=1 and r5job02>0 then do;
    r5self02=-3; r5job02=-3; end;
  if self=0 and r5job02>0 then do;
    r5emp02=-3; r5job02=-3; end;
end;

```

```

end;

array r5job (i) r5job80-r5job99 r5job00-r5job02;
array r5self (i) r5self80-r5self99 r5self00-r5self02;
array r5emp (i) r5emp80-r5emp99 r5emp00-r5emp02;

*** Include valid skips & non-interview cases;
do over r5job;
  if e200=-5 then do;
    r5job=-5; r5self=-5; r5emp=-5;
  end;
end;

array r4job r4job80-r4job99 r4job00 r4job01;
array tjob tjob80-tjob99 tjob00 tjob01;
array r4sjob r4sjob80-r4sjob99 r4sjob00 r4sjob01;
array tsjob tsjob80-tsjob99 tsjob00 tsjob01;
array r4ejob r4ejob80-r4ejob99 r4ejob00 r4ejob01;
array tejob tejob80-tejob99 tejob00 tejob01;
array self self1-self9;
array r4slf r4slf1-r4slf9;

do over tjob;
  r4job=tjob; r4sjob=tsjob; r4ejob=tejob; end;

do over self; r4slf=self; end;

array r1job (i) r1job80-r1job99 r1job00-r1job02;
array r2job (i) r2job80-r2job99 r2job00-r2job02;
array r3job (i) r3job80-r3job99 r3job00-r3job02;
array r4job (i) r4job80-r4job99 r4job00-r4job02;
array r4self (i) r4sjob80-r4sjob99 r4sjob00-r4sjob02;
array r4emp (i) r4ejob80-r4ejob99 r4ejob00-r4ejob02;
array r5job (i) r5job80-r5job99 r5job00-r5job02;
array r5self (i) r5self80-r5self99 r5self00-r5self02;
array r5emp (i) r5emp80-r5emp99 r5emp00-r5emp02;
array tjob (i) tjob80-tjob99 tjob00-tjob02;
array tsjob (i) tsjob80-tsjob99 tsjob00-tsjob02;
array tejob (i) tejob80-tejob99 tejob00-tejob02;

/* Initialize created variable */
do over tjob;
  tjob=0; tsjob=0; tejob=0;
end;

/* Begin by adding together the created variables from both rounds. Jobs that are double counted will be
subtracted off later in the program. */
do over tjob;
  if r5job>0 then do; tjob=r5job; end;
  if r5emp>0 then do; tejob=r5emp; end;
  if r5self>0 then do; tsjob=r5self; end;

```

```

if r4job>0 then do; tjob=r4job; end;
if r4emp>0 then do; tejob=r4emp; end;
if r4self>0 then do; tsjob=r4self;end;
if r3job>0 and r4int=-5 then do;
    tjob=r3job; tejob=r3job;
end;
if r2job>0 and r3int=-5 and r4int=-5 and r3int=-5 then do;
    tjob=r2job; tejob=r2job;
end;
if r1job>0 and r2int=-5 and r3int=-5 and r4int=-5 and intwk>0 then do;
    tjob=r1job; tejob=r1job;
end;

if r4job>0 and r5job>0 then do;
    tjob=r4job+r5job;
end;
if r4emp>0 and r5emp>0 then do;
    tejob=r4emp+r5emp;
end;
if r4self>0 and r5self>0 then do;
    tsjob=r4self+r5self;
end;

if r3job>0 and r5job>0 and r4int=-5 then do;
    tjob=r3job+r5job;
end;
if r3job>0 and r5emp>0 and r4int=-5 then do;
    tejob=r3job+r5emp;
end;

if r2job>0 and r5job>0 and r3int=-5 and r4int=-5 then do;
    tjob=r2job+r5job;
end;
if r2job>0 and r5emp>0 and r3int=-5 and r4int=-5 then do;
    tejob=r2job+r5emp;
end;

if r1job>0 and r5job>0 and r2int=-5 and r3int=-5 and r4int=-5 then do;
    tjob=r1job+r5job;
end;
if r1job>0 and r5emp>0 and r2int=-5 and r3int=-5 and r4int=-5 then do;
    tejob=r1job+r5emp;
end;
end;

/* Define negative values for the total created variable. */
do over tjob;
    if r5job in (-1,-2,-3) then tjob=-3;
    if r5self in (-1,-2,-3) then tsjob=-3;
    if r5emp in (-1,-2,-3) then tejob=-3;
    if r4job in (-1,-2,-3) then tjob=-3;

```

```

if r4self in (-1,-2,-3) then tsjob=-3;
if r4emp in (-1,-2,-3) then tejob=-3;
if r3job in (-1,-2,-3) and r4int=-5 then do;
  tjob=-3;
  tejob=-3;
end;
if r2job in (-1,-2,-3) and r3int=-5 and r4int=-5 and intwk>0 then do;
  tjob=-3; tejob=-3;
end;
if r1job in (-1,-2,-3) and r2int=-5 and r3int=-5 and r4int=-5 and intwk>0 then do;
  tjob=-3; tejob=-3;
end;
end;

array r1uid (i) r1uid1-r1uid9;
array r2uid (i) r2uid1-r2uid9;
array r3uid (i) r3uid1-r3uid9;
array r4uid (i) r4uid1-r4uid9;
array r5uid (i) uid1-uid9;
array r4slf (i) r4slf1-r4slf9;
array self (i) self1-self9;
array match1 (i) match11-match18;
array match2 (i) match21-match28;
array match3 (i) match31-match38;
array match4 (i) match41-match48;
array match5 (i) match51-match58;
array match6 (i) match61-match68;
array match7 (i) match71-match78;
array match8 (i) match81-match88;
array match9 (i) match91-match98;

do i=1 to 8;
  match1=0; match2=0; match3=0; match4=0; match5=0; match6=0; match7=0; match8=0; match9=0;
end;

/* Define match14 as the dummy variable that equals one when the first job on the Round1 UID roster
and the fourth job on the Round2 roster have the same UID. */

/* If any UID from the first position in dli Round matches with any UID in Round 5 */
do over r5uid;
  if r4uid1>0 and r5uid>0 and r4int>0 and intwk>0 then do;
    if r4uid1=r5uid then do; match1=1; end;
  end;
  if r3uid1>0 and r5uid>0 and r4int=-5 and intwk>0 then do;
    if r3uid1=r5uid then do; match1=1; end;
  end;
  if r2uid1>0 and r5uid>0 and r4int=-5 and r3int=-5 and intwk>0 then do;
    if r2uid1=r5uid then do; match1=1; end;
  end;
  if r1uid1>0 and r5uid>0 and r4int=-5 and r3int=-5 and r2int=-5 and intwk>0 then do;
    if r1uid1=r5uid then do; match1=1; end;

```

```
end;  
end;
```

```
/* The above code repeats for UIDs in the second-ninth positions in the dli round. Contact NLS User  
Services for more information. */
```

```
/* Define "same98" as a counter variable that adds up how many matching UIDs are in the respondent's  
UID roster. This will be subtracted from the total number of jobs created variable to avoid double  
counting the same job. Begin by initializing the "same" variables to zero. */  
same97=0;  
same98=0;  
same99=0;  
same00=0;  
same01=0;  
ssame00=0;  
ssame01=0;
```

```
array r1j97 (i) r1j971-r1j979;  
array r1j98 (i) r1j981-r1j989;  
array r2j98 (i) r2j981-r2j989;  
array r2j99 (i) r2j991-r2j999;  
array r3j99 (i) r3j991-r3j999;  
array r3j00 (i) r3j001-r3j009;  
array r4j00 (i) r4j001-r4j009;  
array r4j01 (i) r4j011-r4j019;  
array r5j97 (i) r5j971-r5j979;  
array r5j98 (i) r5j981-r5j989;  
array r5j99 (i) r5j991-r5j999;  
array r5j00 (i) r5j001-r5j009;  
array r5j01 (i) r5j011-r5j019;  
array r5j02 (i) r5j021-r5j029;
```

```
/* Consider the respondents with a Round 1 interview date in 1997, non-interview cases only. */  
if r1int_y=1997 and r2int=-5 and r3int=-5 and r4int=-5 and intwk>0 then do;
```

```
do over match1;  
  if match1=1 and r5j97=1 and r1j971=1 then do;  
    same97=same97+1;  
  end;  
end;  
do over match2;  
  if match2=1 and r5j97=1 and r1j972=1 then do;  
    same97=same97+1;  
  end;  
end;  
/*and so on through */  
do over match9;  
  if match9=1 and r5j97=1 and r1j979=1 then do;  
    same97=same97+1;  
  end;  
end;  
end;
```

```

/* Consider the respondents with a Round 1 interview date in 1998, non-interview cases only. */
if r1int_y=1998 and r2int=-5 and r3int=-5 and r4int=-5 and intwk>0 then do;
do over match1;
  if match1=1 and r5j98=1 and r1j981=1 then do;
    same98=same98+1;
  end;
end;
do over match2;
  if match2=1 and r5j98=1 and r1j982=1 then do;
    same98=same98+1;
  end;
end;
/*and so on through */
do over match9;
  if match9=1 and r5j98=1 and r1j989=1 then do;
    same98=same98+1;
  end;
end;
end;

/* Consider the respondents with a Round 2 interview date in 1998. */
if r2int_y=1998 and r3int=-5 and r4int=-5 and intwk>0 then do;
do over match1;
  if match1=1 and r5j98=1 and r2j981=1 then do;
    same98=same98+1;
  end;
end;
do over match2;
  if match2=1 and r5j98=1 and r2j982=1 then do;
    same98=same98+1;
  end;
end;
/*and so on through */
do over match9;
  if match9=1 and r5j98=1 and r2j989=1 then do;
    same98=same98+1;
  end;
end;
end;

/* Consider the respondents with a Round 2 interview date in 1999. */
if r2int_y=1999 and r3int=-5 and r4int=-5 and intwk>0 then do;
do over match1;
  if match1=1 and r5j99=1 and r2j991=1 then do;
    same99=same99+1;
  end;
end;
do over match2;
  if match2=1 and r5j99=1 and r2j992=1 then do;
    same99=same99+1;
  end;
end;

```

```

    end;
end;
/*and so on through */
do over match9;
  if match9=1 and r5j99=1 and r2j999=1 then do;
    same99=same99+1;
  end;
end;
end;

/* Consider the respondents with a Round 3 interview date in 1999. */
if r3int_y=1999 and r4int=-5 and intwk>0 then do;
  do over match1;
    if match1=1 and r5j99=1 and r3j991=1 then do;
      same99=same99+1;
    end;
  end;
  do over match2;
    if match2=1 and r5j99=1 and r3j992=1 then do;
      same99=same99+1;
    end;
  end;
end;
/*and so on through */
do over match9;
  if match9=1 and r5j99=1 and r3j999=1 then do;
    same99=same99+1;
  end;
end;
end;

/* Consider the respondents with a Round 3 interview date in 2000. */
if r3int_y=2000 and r4int=-5 and intwk>0 then do;
  do over match1;
    if match1=1 and r5j00=1 and r3j001=1 then do;
      same00=same00+1;
    end;
  end;
  do over match2;
    if match2=1 and r5j00=1 and r3j002=1 then do;
      same00=same00+1;
    end;
  end;
end;
/*and so on through */
do over match9;
  if match9=1 and r5j00=1 and r3j009=1 then do;
    same00=same00+1;
  end;
end;
end;

/* Consider the respondents with a Round 4 interview date in 2000. */

```

```

/* "ssame" refers to a match of self-employed jobs, only needed for Round 4 since there were no self-employed jobs on the roster before Round 4. */

if (r4int_y=2000 and intwk>0) or back4=1 then do;
  do over match1;
    if match1=1 and r5j00=1 and r4j001=1 then do;
      same00=same00+1;
    end;
    if match1=1 and r5j00=1 and r4j001=1 and self=1 and r4slf1=1 then do;
      ssame00=ssame00+1;
    end;
  end;
do over match2;
  if match2=1 and r5j00=1 and r4j002=1 then do;
    same00=same00+1;
  end;
  if match2=1 and r5j00=1 and r4j002=1 and self=1 and r4slf2=1 then do;
    ssame00=ssame00+1;
  end;
end;
/*and so on through */
do over match9;
  if match9=1 and r5j00=1 and r4j009=1 then do;
    same00=same00+1;
  end;
  if match9=1 and r5j00=1 and r4j009=1 and self=1 and r4slf9=1 then do;
    ssame00=ssame00+1;
  end;
end;
/* Consider the respondents with a Round 3 interview date in 2000. */
if r4int_y=2001 and intwk>0 then do;
  do over match1;
    if match1=1 and r5j01=1 and r4j011=1 then do;
      same01=same01+1;
    end;
    if match1=1 and r5j01=1 and r4j011=1 and self=1 and r4slf1=1 then do;
      ssame01=ssame01+1;
    end;
  end;
do over match2;
  if match2=1 and r5j01=1 and r4j012=1 then do;
    same01=same01+1;
  end;
  if match2=1 and r5j01=1 and r4j012=1 and self=1 and r4slf2=1 then do;
    ssame01=ssame01+1;
  end;
end;
/*and so on through */
do over match9;

```

```

if match9=1 and r5j01=1 and r4j019=1 then do;
    same01=same01+1;
end;
if match9=1 and r5j01=1 and r4j019=1 and self=1 and r4slf9=1 then do;
    ssame01=ssame01+1;
end;
end;
end;

```

/\* By construction of the same variable, employee-type same must be the difference between total and self-employed same. This variable is only necessary for Round 4 (years 00 and 01) since self-employed jobs were not on the roster until Round 4. \*/

```

esame00=0;
esame01=0;

```

```

esame00=same00-ssame00;
esame01=same01-ssame01;

```

/\* Now subtract the "same" count variable from the total created variable computed above. \*/

```

if tjob97=>0 then do; tjob97=tjob97-same97; end;
if tejob97=>0 then do; tejob97=tejob97-same97; end;
if tjob98=>0 then do; tjob98=tjob98-same98; end;
if tejob98=>0 then do; tejob98=tejob98-same98; end;
if tjob99=>0 then do; tjob99=tjob99-same99; end;
if tejob99=>0 then do; tejob99=tejob99-same99; end;
if tjob00=>0 then do; tjob00=tjob00-same00; end;
if tejob00=>0 then do; tejob00=tejob00-esame00; end;
if tsjob00=>0 then do; tsjob00=tsjob00-ssame00; end;
if tjob01=>0 then do; tjob01=tjob01-same01; end;
if tejob01=>0 then do; tejob01=tejob01-esame01; end;
if tsjob01=>0 then do; tsjob01=tsjob01-ssame01; end;

```

/\* Non-interview cases \*/

```

do over tjob;
    if uid1=-5 then do;
        tjob=-5; tejob=-5; tsjob=-5;
    end;
end;
endsas;

```

## Total Number of Jobs Held Age 14-19, Number of Employee Jobs Held since Age 20, and Number of All Jobs Held since Age 20

**Variables Created:** CV\_TTL\_JOBS\_TEEN  
                   CV\_TTL\_JOBS\_ADULT\_ET  
                   CV\_TTL\_JOBS\_ADULT\_ALL

## Programs Used

This program uses [emp\\_begin.sas](#) as input (see the first page of this appendix for details).

This program calculates the total number of employee-type jobs held by the respondent between the ages of 14 and 19. It then creates two variables for the total number of jobs held by the respondent since age 20: one for all jobs and one for employee-type jobs. See the [introduction to this appendix for important information about the universes for each of these variables](#). The variables are only created for respondents who have worked at least one week in the applicable age range.

---

```
array job1wks (i) wk1_1-wk1_1200;
array job3wks (i) wk3_1-wk3_1200;
array job5wks (i) wk5_1-wk5_1200;
array job7wks (i) wk7_1-wk7_1200;
array starfl (i) starfl_1-starfl_8;           array job2wks (i) wk2_1-wk2_1200;
array job4wks (i) wk4_1-wk4_1200;
array job6wks (i) wk6_1-wk6_1200;
array job8wks (i) wk8_1-wk8_1200;
array stopfl (i) stopfl_1-stopfl_8;

** Indicate if worked at least one week on a given job since age 14 & 20 **;

job141=0;  job201=0;  ejob201=0;          job142=0;  job202=0;  ejob202=0;
job143=0;  job203=0;  ejob203=0;          job144=0;  job204=0;  ejob204=0;
job145=0;  job205=0;  ejob205=0;          job146=0;  job206=0;  ejob206=0;
job147=0;  job207=0;  ejob207=0;          job148=0;  job208=0;  ejob208=0;

/* Respondents under 20 at interview date, count jobs worked to date. */
if age14wk>0 and (age20wk-1)>1200 then do;
  do i=age14wk to 1200;
    if job1wks in (-3,1) and self1=0 then do; job141=1; end;
    if job2wks in (-3,1) and self2=0 then do; job142=1; end;
    /* and so on through */
    if job8wks in (-3,1) and self8=0 then do; job148=1; end;
  end;
end;

/* Respondents over 20 at interview date, count jobs worked before age 20. */
if age14wk>0 and (age20wk-1)<=1200 then do;
  do i=age14wk to (age20wk-1);
    if job1wks in (-3,1) and self1=0 then do; job141=1; end;
    if job2wks in (-3,1) and self2=0 then do; job142=1; end;
    /* and so on through */
    if job8wks in (-3,1) and self8=0 then do; job148=1; end;
  end;
end;

/* Respondents over 20 at interview date, count employee jobs worked after age 20. */
if age14wk>0 and (age20wk-1)<=1200 then do;
  do i=age20wk to 1200;
    if job1wks in (-3,1) and self1=0 then do; ejob201=1; end;
    if job2wks in (-3,1) and self2=0 then do; ejob202=1; end;
    /* and so on through */
    if job8wks in (-3,1) and self8=0 then do; ejob208=1; end;
  end;
end;
```

```

    end;
end;

/* Respondents over 20 at interview date, count jobs worked after age 20. */
if age14wk>0 and (age20wk-1)<=1200 then do;
  do i=age20wk to 1200;
    if job1wks in (-3,1) then do; job201=1; end;
    if job2wks in (-3,1) then do; job202=1; end;
    /* and so on through */
    if job8wks in (-3,1) then do; job208=1; end;
  end;
end;

```

```

r5emp14=sum(job141,job142,job143,job144,job145,job146,job147,job148);
r5job20=sum(job201,job202,job203,job204,job205,job206,job207,job208);
r5emp20=sum(ejob201,ejob202,ejob203,ejob204,ejob205,ejob206,ejob207,ejob208);

```

\* only need to assign an invalid skip to cases where the stop date is unknown and the job starts prior to the 14/20<sup>th</sup> birthday;

```

if stopfl_1=1 and starfl_1 ne 1 and starw1<age14wk and self1=0 then do; r5emp14=-3; end;
if stopfl_1=1 and starfl_1 ne 1 and starw1<age20wk and self1=0 and age20wk<intwk then do;
  r5emp20=-3; end;
if stopfl_1=1 and starfl_1 ne 1 and starw1<age20wk and age20wk<intwk then do; r5job20=-3; end;
if stopfl_2=1 and starfl_2 ne 1 and starw2<age14wk and self2=0 then do; r5emp14=-3; end;
if stopfl_2=1 and starfl_2 ne 1 and starw2<age20wk and self2=0 and age20wk<intwk then do;
  r5emp20=-3; end;
if stopfl_2=1 and starfl_2 ne 1 and starw2<age20wk and age20wk<intwk then do; r5job20=-3; end;
if stopfl_3=1 and starfl_3 ne 1 and starw3<age14wk and self3=0 then do; r5emp14=-3; end;
if stopfl_3=1 and starfl_3 ne 1 and starw3<age20wk and self3=0 and age20wk<intwk then do;
  r5emp20=-3; end;
if stopfl_3=1 and starfl_3 ne 1 and starw3<age20wk and age20wk<intwk then do; r5job20=-3; end;
if stopfl_4=1 and starfl_4 ne 1 and starw4<age14wk and self4=0 then do; r5emp14=-3; end;
if stopfl_4=1 and starfl_4 ne 1 and starw4<age20wk and self4=0 and age20wk<intwk then do;
  r5emp20=-3; end;
if stopfl_4=1 and starfl_4 ne 1 and starw4<age20wk and age20wk<intwk then do; r5job20=-3; end;
if stopfl_5=1 and starfl_5 ne 1 and starw5<age14wk and self5=0 then do; r5emp14=-3; end;
if stopfl_5=1 and starfl_5 ne 1 and starw5<age20wk and self5=0 and age20wk<intwk then do;
  r5emp20=-3; end;
if stopfl_5=1 and starfl_5 ne 1 and starw5<age20wk and age20wk<intwk then do; r5job20=-3; end;
if stopfl_6=1 and starfl_6 ne 1 and starw6<age14wk and self6=0 then do; r5emp14=-3; end;
if stopfl_6=1 and starfl_6 ne 1 and starw6<age20wk and self6=0 and age20wk<intwk then do;
  r5emp20=-3; end;
if stopfl_6=1 and starfl_6 ne 1 and starw6<age20wk and age20wk<intwk then do; r5job20=-3; end;
if stopfl_7=1 and starfl_7 ne 1 and starw7<age14wk and self7=0 then do; r5emp14=-3; end;
if stopfl_7=1 and starfl_7 ne 1 and starw7<age20wk and self7=0 and age20wk<intwk then do;
  r5emp20=-3; end;
if stopfl_7=1 and starfl_7 ne 1 and starw7<age20wk and age20wk<intwk then do; r5job20=-3; end;
if stopfl_8=1 and starfl_8 ne 1 and starw8<age14wk and self8=0 then do; r5emp14=-3; end;
if stopfl_8=1 and starfl_8 ne 1 and starw8<age20wk and self8=0 and age20wk<intwk then do;
  r5emp20=-3; end;
if stopfl_8=1 and starfl_8 ne 1 and starw8<age20wk and age20wk<intwk then do; r5job20=-3; end;

```

```

if e200=-5 then do;
  r5emp14=-5; r5job20=-5; r5emp20=-5;
end;

array job14 (i) job141-job149;
array job20 (i) job201-job209;
array r4j14 (i) r4j141-r4j149;
array r4j20 (i) r4j201-r4j209;

array self (i) self1-self9;
array r4self (i) r4self1-r4self9;

do over job14; r4j14=job14; r4j20=job20; end;
do over self; r4self=self; end;

r4emp14=totjob14;
r4job20=totjob20;
r4emp20=totemp20;

/* Initialize final created variable */
totemp14=0;
totjob20=0;
totemp20=0;

/* Begin by adding together the created variables from both rounds. Jobs that are double counted will be
subtracted off later in the program. */
if r5emp14>0 then do; totemp14=r5emp14; end;
if r4emp14>0 then do; totemp14=r4emp14; end;
if r3job14>0 and r4int=-5 then do; totemp14=r3job14; end;
if r2job14>0 and r3int=-5 and r4int=-5 and intwk>0 then do; totemp14=r2job14; end;
if r1job14>0 and r4int=-5 and r3int=-5 and r2int=-5 and intwk>0 then do; totemp14=r1job14; end;

if r4emp14>0 and r5emp14>0 then do; totemp14=r4emp14+r5emp14; end;
if r3job14>0 and r5emp14>0 and r4int=-5 then do; totemp14=r3job14+r5emp14; end;
if r2job14>0 and r5emp14>0 and r3int=-5 and r4int=-5 then do; totemp14=r2job14+r5emp14; end;
if r1job14>0 and r5emp14>0 and r2int=-5 and r3int=-5 and r4int=-5 then do;
  totemp14=r1job14+r5emp14; end;

if r5job20>0 then do; totjob20=r5job20; end;
if r4job20>0 then do; totjob20=r4job20; end;
if r3job20>0 and r4int=-5 then do; totjob20=r3job20; end;

if r4job20>0 and r5job20>0 then do; totjob20=r4job20+r5job20; end;
if r3job20>0 and r5job20>0 and r4int=-5 then do; totjob20=r3job20+r5job20; end;

if r5emp20>0 then do; totemp20=r5emp20; end;
if r4emp20>0 then do; totemp20=r4emp20; end;
if r3job20>0 and r4int=-5 then do; totemp20=r3job20; end;
if r4emp20>0 and r5emp20>0 then do; totemp20=r4emp20+r5emp20; end;
if r3job20>0 and r5emp20>0 and r4int=-5 then do; totemp20=r3job20+r5emp20; end;

```

```

array r4j14 (i) r4j141-r4j149;
array r4j20 (i) r4j201-r4j209;
array job14 (i) job141-job149;
array job20 (i) job201-job209;
array r1uid (i) r1uid1-r1uid7;
array r2uid (i) r2uid1-r2uid9;
array r3uid (i) r3uid1-r3uid9;
array r4uid (i) r4uid1-r4uid9;
array r5uid (i) uid1-uid9;
array match1 (i) match11-match18;
array match2 (i) match21-match28;
array match3 (i) match31-match38;
array match4 (i) match41-match48;
array match5 (i) match51-match58;
array match6 (i) match61-match68;
array match7 (i) match71-match78;
array match8 (i) match81-match88;
array match9 (i) match91-match98;

do i=1 to 8;
  match1=0; match2=0; match3=0; match4=0; match5=0; match6=0; match7=0; match8=0; match9=0;
end;

/* Define match14 as the dummy variable that equals one when the first job on the Round1 UID roster
and the fourth job on the Round2 roster have the same UID. */

/* If any UID from the first position in dli Round matches with any UID in Round 5 */
do over r5uid;
  if r4uid1>0 and r5uid>0 and r4int>0 and intwk>0 then do;
    if r4uid1=r5uid then do; match1=1; end;
    end;
  if r3uid1>0 and r5uid>0 and r4int=-5 and intwk>0 then do;
    if r3uid1=r5uid then do; match1=1; end;
    end;
  if r2uid1>0 and r5uid>0 and r4int=-5 and r3int=-5 and intwk>0 then do;
    if r2uid1=r5uid then do; match1=1; end;
    end;
  if r1uid1>0 and r5uid>0 and r4int=-5 and r3int=-5 and r2int=-5 and intwk>0 then do;
    if r1uid1=r5uid then do; match1=1; end;
    end;
  end;

/* The above code repeats for UIDs in the second-ninth positions in the dli round. Contact NLS User
Services for more information. */

/* Define "same" as a counted variable that adds up how many matching UIDs are in the respondent's
UID roster. This will be subtracted from the total number of jobs created variable to avoid double
counting the same job. Begin by initializing the "same" variables to zero. */
same14=0;
same20=0;

```

```

esame20=0;

array r4self (i) r4self1-r4self9;
array self (i) self1-self9;

do over match1;
  /* dli is Round 4 */
  if match1=1 and r4j141=1 and job14=1 and r4self1=0 and self=0 then do; same14=same14+1; end;
  if match1=1 and r4j201=1 and job20=1 then do; same20=same20+1; end;

  /* Creation of same variable for employee-type jobs only.*/
  if match1=1 and r4j201=1 and job20=1 and r4self1=0 and self=0 then do; esame20=esame20+1; end;

  /* dli is round 3, no need for to account for self emp. jobs because all jobs Round 3 and earlier are
employee-type.*/
  if match1=1 and r3j141=1 and job14=1 and r4int=-5 then do; same14=same14+1; end;
  if match1=1 and r4j201=1 and job20=1 and r4int=-5 then do; same20=same20+1; end;

  /* dli is Round 2. No need for same20 here since everyone in the sample was under 20 during Rounds
1 and 2.*/
  if match1=1 and job14=1 and r4int=-5 and r3int=-5 then do; same14=same14+1; end;

  /* dli is Round 1*/
  if match1=1 and job14=1 and r4int=-5 and r3int=-5 and r2int=-5 then do; same14=same14+1; end;
end;

/* The above code repeats for match2 - match9. Contact NLS User Services for information. */

/* Now subtract the "same" count variable from the total created variable computed above.*/
if totemp14=>0 then do; totemp14=totemp14-same14; end;
if totjob20=>0 then do; totjob20=totjob20-same20; end;
if totemp20=>0 then do; totemp20=totemp20-esame20; end;

/* Define negative values for the total created variable.*/
if r1job14 in (-1,-2,-3) and intwk>0 and r4int=-5 and r3int=-5 and r2int=-5 then totemp14=-3;
if r2job14 in (-1,-2,-3) and intwk>0 and r4int=-5 and r3int=-5 then totemp14=-3;
if r3job14 in (-1,-2,-3) and intwk>0 and r4int=-5 then totemp14=-3;
if r4emp14 in (-1,-2,-3) then totemp14=-3;
if r5emp14 in (-1,-2,-3) then totemp14=-3;

if r3job20 in (-1,-2,-3) and r4int=-5 and intwk>0 then totjob20=-3;
if r4job20 in (-1,-2,-3) then totjob20=-3;
if r5job20 in (-1,-2,-3) then totjob20=-3;

if r3job20 in (-1,-2,-3) and r4int=-5 and intwk>0 then totemp20=-3;
if r4emp20 in (-1,-2,-3) then totemp20=-3;
if r5emp20 in (-1,-2,-3) then totemp20=-3;

/* For respondents younger than 20 at interview date, their thrs14 c.v. will be -4.*/
if age20wk>intwk and intwk>0 then do; totjob20=-4; totemp20=-4; end;

```

```
if uid1=-5 then do;
  totemp14=-5; totjob20=-5; totemp20=-5;
end;

endsas;
```

---

## Job Length Less Than 13 Weeks

**Variables Created:** CV\_JOB<13\_WKS

### Variables Used

E200A	YEMP-200A
E377011-E377017	YEMP-37701.01-.07
E37901B1-E37901B7	YEMP-37901B.01-.07
E582011-E582018	YEMP-58201.01-.08
E599001-E599008	YEMP-59900.01-.08
E584011-E584018	YEMP-58401.01-.08
E584031-E584037	YEMP-58403.01-.07
PUBID	PUBID

This program creates a variable indicating whether each of the respondent's jobs lasted more than 13 weeks.

---

```
***** Define the arrays for the variables used;
array e37701 e377011-e377019;
array e58201 e582011-e582019;
array e37901b e37901b1-e37901b9;
array e59900 e599001-e599009;
array e58401 e584011-e584019;
```

```
***** Define the arrays for the created variables;
array jleng jleng01-jleng09;
```

```
***** initialize the variables;
do i=1 to 9;
  jleng[i]=-4;
end;
```

```
***** Create the variables;
do i=1 to 9;
  if e37701[i]=0 or e37901b[i]=0 then jleng[i]=1;
  else if e58201[i]=0 or e59900[i]=0 then jleng[i]=1;
  else if e58401[i]=1 then jleng[i]=1;
  else if e37901b[i]=1 or e59900[i]=1 then jleng[i]=0;
  else if e37901b[i] in (-1,-2,-3) or e59900[i] in (-1,-2,-3) then jleng[i]=-3;
```

```
/* Since people who answer e58401 already answer e37701(e37901b) or e58201(e59900), don't assign -  
3 based on the invalid answer to e58401. */  
else if e200a=-5 then jleng[i]=-5;  
end;  
  
endsas;
```

---